IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 12, 2021, accepted July 21, 2021, date of publication July 26, 2021, date of current version August 4, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3100141

F/Wvis: Hierarchical Visual Approach for Effective Optimization of Firewall Policy

TAEYONG KIM, TAEWOONG KWON, JUN LEE, AND JUNGSUK SONG
Science and Technology Cyber Security Center, Korea Institute of Science and Technology Information (KISTI), Daejeon 34141, South Korea
Corresponding author: Jungsuk Song (song@kisti.re.kr)

This work was supported by 'Construction of Information Security Scheme for Supercomputing Environment based on AI (K-21-L02-C03)', funded by Korea Institute of Science and Technology Information (KISTI).

ABSTRACT As an essential system for protecting internal networks and valuable information, the firewall monitors and controls network traffic in terms of access control, authentication, logging, and auditing. In particular, it carries out both allowing and blocking communications between internal and external networks based on proper Access Control List (ACL). However, a complex ACL along with huge network environments lead to exposing vulnerabilities and communication problems, because of anomalies among policies. Even though various techniques and applications combined with visualization approaches have been proposed, there is still a lack of usability caused by not only the limitation of the text-based interface but also the complexity of practical use. In order to solve these problems, this work proposes a 3D-based hierarchical visualization method, namely F/Wvis, for intuitive ACL management and analysis. The F/Wvis, particularly, supports ACL management for a large-scale network as well as analysis of detail anomalies on policies by providing a drill-down user interface through the hierarchical visualization approach. Further, the implemented system is evaluated against popular tools by network security experts to identify the usability and effectiveness in real-world situations (a demonstration video is available at: https://bit.ly/34ooEDc).

INDEX TERMS Firewall, information visualization, policy anomaly, user interface, usability.

I. INTRODUCTION

Firewall is being widely used as a fundamental element of network security. In order to control network communications securely, the firewall access control list consisting of policies (i.e., rules) manages access from external networks as well as transmission of packets departing from internal networks. Even though various devices (e.g., Intrusion Detection/Prevention System (IDS/IPS) or Security Information Event Management(SIEM)) have been developed to protect network resources nowadays, firewall is still considered as the most priority device from network and security administrators in real-world companies [1].

To control network traffic, firewall inspects every packet using policies which consist of a set of tuples, such as IPs (i.e., both source and destination), port (mostly destination), protocols (e.g., Transmission Control Protocol (TCP)), User Datagram Protocol (UDP), and so on), and actions (i.e., allow and deny); comparing these factors with in/out packets, firewall tries to allow or deny ingress traffic to, and egress traffic from internal network. Firewall policies are typically established based on 'ANY-ANY-DENY' operation which blocks all traffic at the first step, after which it allows some policies such as *whitelisting*; whereas, 'ANY-ANY-ALLOW' which is mainly used for communications on the internal network, requires to pay particular attention to administrators since it will permit all traffic [2]–[4]. The achievement of high levels of security can be guaranteed based on proper policies which are configured by a proficient administrator. With widespread of network devices and abnormal activities, however, the number of firewall policies has increased rapidly; accordingly it can incur unexpected anomalies between policies. In fact, many studies make an effort to reveal anomalies, since outdated firewall policies may provoke unauthorized network access and cyber attacks [5]–[7].

The firewall policy should be handled with great care, because a single anomaly can put entire network under a

The associate editor coordinating the review of this manuscript and approving it for publication was Sedat Akleylek.

security risk. However, dozens or hundreds of policies make it difficult to the firewall administrator with regard to the policy management and optimization. Furthermore, since policies are commonly managed as a text-based tabular representation from popular firewall management systems, the administrator could not figure out lots of policies intuitively with current situations. Moreover, the sequence of policies and the relationship between entire policies are required to consider for maintenance of large-scale of policies (i.e., multiple ACLs). To solve this, various tools based on visual approaches have proposed to improve the readability and usability for actual use in complex real-world situations [8]–[11]. Nevertheless, it is hard to not only become well-acquainted with the visualization systems for a user, but come up with an elaborate visualization of complex policies using two-dimensional simple graphs.

In this paper, we propose a novel hierarchical visual approach to optimizing the ACL of firewall on large-scale networks. The proposed method analyzes policy anomalies in the entire ACL policies automatically; afterwards it visualizes a firewall information to the three-dimensional (3D) space revolving around anomalies between policies, considering a connectivity and usage of policies. Particularly, we define two layers to represent a summary information of anomalies (i.e., upper layer) and provide details of policies (i.e., lower layer) as a linked graph structure. Thus, the visualization provides anomaly information depending on the order of policies as well as helps to revise anomaly policies with an intuitive drill-down search interface. In order to identify effectiveness and usability of the method, we carry out comprehensive evaluations of both quantitative analysis of the performance and quantitative assessment from network experts using large-scale, real-world firewall policy data. The main contributions of this paper are three-fold:

- **Hierarchical visual approach**: We propose a novel visual approach to provide intuitive assistance in the firewall policy management. The policies and their anomalies are visualized as two layers in terms of both the overall summary (upper layer) and detail relations (lower layer) to resolve a complex policy anomaly (Section III).
- **Real-world feasibility**: We implement the concept of the methodology to the three-dimensional space with the interactive user interface, and present a feasibility case study involving management of a real-world large-scale firewall policies (e.g., a data has more than 2,000 terminals, 300 policies, and involves 100 anomalies) (Section IV).
- **Comprehensive evaluation**: The effectiveness of the method is validated by elaborate experiments based on not only various real-world data sets (six organizations), but also high quality human resource (six experts); considering for practical use. The result shows outstanding performance and reduction of resolution time (less than tenfold) to solve anomalies compared with popular tools (Section V).

This paper is organized as follows. Section 2 shortly reviews related work encompassing the roles of firewall and its policy, types of policy anomalies, and popular visual tools for policy management. In Section 3, we present our methodology for the three-dimensional hierarchical visualization with the drill-down interface, followed by the system implementation with practical use scenarios in Section 4. Section 5 evaluates the methodology in terms of quantitative/quantitative analysis. Finally, the discussion of our findings and limitations and the conclusions are presented in Section 6 and 7, respectively.

II. RELATED WORK

To understand the background related to the firewall policy and its management, we look into important considerations, such as the role and necessity of firewall with access policy, anomalies which lead to network problems, and tools for management of the access policy bringing together a visualization method.

A. THE ROLES OF FIREWALL AND ACCESS POLICY

The operation of the firewall mainly employs the packet filtering which allows network packets to pass or halt by predefined policies based on the source and destination Internet Protocol (IP) addresses, protocols and ports. Comparing packets to the policy is a simple, but effective way to control security levels [12], [13]. However, there have been some concerns surrounding the degradation in large network performance of the filters in proportion to the number of policies. Moreover, it happens to be the case that the firewall allows every port for supporting a specific service, such as File Transfer Protocol (FTP) involved derived sessions; in this case, it might cause serious security incidents.

To enhance the performance of firewall, the Stateful Packet Inspection (SPI) firewall, as an improved version of the packet filtering, identifies a packet's "*state*" based on all connections passing through the firewall [14]–[16]. Nowadays, the integrated security solution tries to contain the role of firewall with advanced functions; for example, Web Application Firewall (WAF) for filtering the content of specific web applications [17], Intrusion Prevention System (IPS) for handling unveiled threats having a normal traffic form [18], and Unified Threat Management (UTM) for providing multiple security functions from single hardware or software as an all-in-one approach [19].

Even though advanced techniques of the firewall are developed for better network security, the packet filtering configured from policies of ACL is, by far, the most effective way to the full or partial function of the firewall. The policy is defined filtering fields, such as IP, port, protocols, and actions and fulfills a one-to-one comparison between every field and in/out packet [20], [21]. In particular firewall policies have a priority order that determines the sequence in which the policies are applied to network traffic. Therefore, the management (e.g., insertion, removal, and modification) of policies requires clear understanding and thorough evaluation of the

relationship between policies; however, it is a big challenge for large-scale networks, considering quite a lot of network devices and policies [22]–[24].

$$P = \{p_0, p_1, \ldots, p_{n-1}, p_n\}, (p_i \mid p_i \in P) \quad (1)$$

$$p_i = \{c_i, a_i\}, (i \mid 0 \leq i \leq n) \quad (2)$$

In fact, the policy management, especially for enterprise networks, has become complex and error-prone; accordingly, periodic checks and quick updates of the policies are needed to carry out for the network security [25]. Unfortunately, it requires the advanced level of proficiency and time-consuming analysis to the network administrators, even though recent studies have been trying to resolve errors among polices and optimize a performance of firewall [26]–[28]. For this reason, it is important to reveal policy conflicts and potential problems automatically, and provide an intuitive solution to the network administrator. We will discuss tools and visual methods for the management of firewall policy later (Section II-C) in more detail.

B. DISCOVERY OF POLICY ANOMALIES

From the early studies of the firewall policy [5], [24], [29]–[32], the anomaly of policy was commonly defined and categorized as four types: `shadowing` (3), `redundancy` (4), `correlation` (5) and `generalization` (6):

$$(\forall f,\ (P_a^f \supseteq P_b^f)) \wedge (P_a^{action} \neq P_b^{action}) \quad (3)$$

$$(\forall f,\ (P_a^f = P_b^f)) \wedge (P_a^{action} = P_b^{action}) \quad (4)$$

$$(\exists f,\ (P_a^f \supseteq P_b^f)) \wedge (\exists f',\ (P_a^{f'} \subseteq P_b^{f'})) \wedge (P_a^{action} \neq P_b^{action}) \quad (5)$$

$$(\forall f,\ (P_a^f \subseteq P_b^f)) \wedge (P_a^{action} \neq P_b^{action}) \quad (6)$$

where P is policy, a and b are orders of the policies such as $a > b$, *action* is the operation of firewall, namely *Allow* or *Deny*, f is the individual fields of a policy, i.e., $f \in \{protocol, s\_IP, s\_port, d\_IP, d\_port\}$, and always $f \neq f'$.

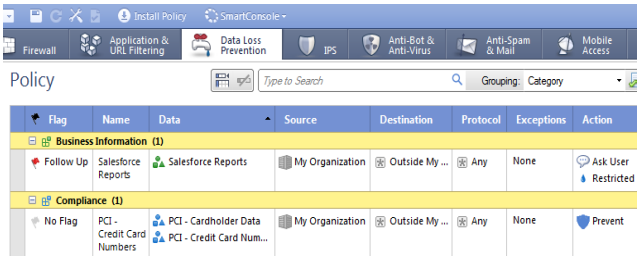
These anomalies include clear conflicts that cause some policies to be always suppressed by other policies, or warnings for potential conflicts [33]; in more detail, the policy *P2* in Table 1 could be shadowed by *P1*, because the previous policy (*P1*) matches all the fields of *P2* with directly opposed actions. In this case, *P2* will never be evaluated, wherefore the shadowing might cause a critical error depend on the priority of policy. Besides the shadowing, *P5* and *P6* specified different actions (i.e., Deny and Allow, respectively) and some fields (e.g., protocol: TCP, s_port: ANY, d_IP: 192.168.3.*, and d_port: 80) that satisfy *P5* also satisfy *P6* and vice-versa; in other words, the correlation anomaly. Moreover, there is a generalization between *P5* and *P6*, because *P6* can process all packets that were handled by *P5* but they take different actions. In addition, *P3* and *P4* are definitely redundant, since they specified the same action with the same conditions of individual fields. The anomaly within policies can make a network in a halting state or lead to a system problem; so the policy management requires careful consideration, even not only inserting a new policy, but deleting and modifying an existing policy.

TABLE 1. An example of firewall policy.

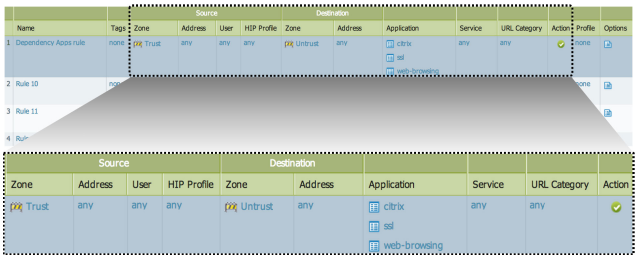
ID	*protocol*	*s_IP*	*s_port*	*d_IP*	*d_port*	*action*
P1	TCP	140.192.34.*	ANY	192.168.1.12	80	Allow
P2	TCP	140.192.34.5	ANY	192.168.1.12	80	Deny
P3	TCP	192.168.1.1	ANY	140.192.33.40	80	Allow
P4	TCP	192.168.1.1	ANY	140.192.33.40	80	Allow
P5	TCP	192.168.3.40	ANY	192.168.3.*	80	Deny
P6	TCP	*.*.*.*	ANY	192.168.3.*	80	Allow

C. VISUAL APPROACHES FOR FIREWALL POLICY MANAGEMENT

To manage complex firewall policies, various approaches and tools have been proposed, especially for the configuration of real-world networks from the firewall manufacturers and vendors [34]–[36]. Although they developed and provided their own tools including a user interface, the approaches are still not so different as most of them adopt a tabular representation, as shown in Figure 1. Given an ACL with a small number of policies, the table format is obviously helpful to manage the list; since firewall applies the policy at the top of the ACL in sequential order, the administrator simply performs all pairwise comparisons for the policies.



(a) Checkpoint firewall [34].



(b) Paloalto firewall [35].

FIGURE 1. Text-based tabular user interfaces for policy management. (private data (e.g., real IP, application, and service) was blinded on (b)).

Nowadays, as the network grows, a visualization approach is exploited to provide an intuitive understanding of the complex policy relations and help the management of a large-scale ACL. From the earlier studies of the policy visualization [8], [37], the policy was expressed as a graph on the two-dimensional surface based on the fields

(usually IP and port). In addition to these simple visual approaches, the interactive functionalities were combined with the visual user interface to enhance the visual inspection for policy anomalies; for instance, F. Mansmann *et al.* [10] proposed a hierarchical sunburst visualization with a color-linked configuration editor and tree view components. NViZ [9] tried to connect policies which communicate with each other, and visualized the graph integrated with important traffic information.

The visual approaches greatly improved legibility and visibility over the complex firewall policies, in comparison with the text-based tabular interface. These systems give us not only colourful views, but intuition that is difficult to recognize in a text-based interface [11], [38]. However, they still have two limitations to use in practical environments. The one is visibility. When visually expressing a narrow range of IP, it showed excellent visibility. But expressing a wide range (2^{32}) of IP as an image, there is a disadvantage that the image is too small to distinguish anomalies. Most of the firewall policy visualization systems have used *grid* or *Venn diagram* method for expressing the anomalies they have [31], [32], [39]–[41]. It requires sufficient space to express all ranges of IP and the overlapping parts but most of us cannot have it. And then, the other is sacrifice of information. In order to improve visibility, some information is sacrificed or distorted or hided to enhance the administrator's intuition. However, it has to be minimized because it is a direct cause of an increase in analysis time, such as performing additional verification procedures. Finally, existing solutions and methodologies can't predict new anomalies that may occur when the policies are modified to resolve existing anomalies. It is a very big disadvantage because it can cause another confusion. To solve these three limitations of preventing time-consuming analysis such as a repetitive search and modification operation, we propose a novel hierarchical visual approach in this paper. It helps to improve intuition for rapidly understanding implicit anomalies in firewall policies and solving them.

III. METHODOLOGY

A. ARCHITECTURE OF HIERARCHICAL VISUAL APPROACH

The realistic reasons, why it is difficult and costly to solve anomalies by popular (i.e., text-based) systems, are as follows: (ℜ 1) difficulty of recognizing anomalies, (ℜ 2) struggling of establishing the cause (policy), (ℜ 3) hardship of determining the priority of policies (when inserting, deleting, and modifying anomaly policies), and (ℜ 4) challenge of estimating derived anomalies (from (ℜ 3)).

To remedy the drawbacks and visualize a complex ACL *at a glance*, this study proposes the hierarchical visualization method with the drill-down user interface for mitigation of policy complexity, reducing time-consuming, and managing error-prone policy to the administrator as follows (see Figure 2):

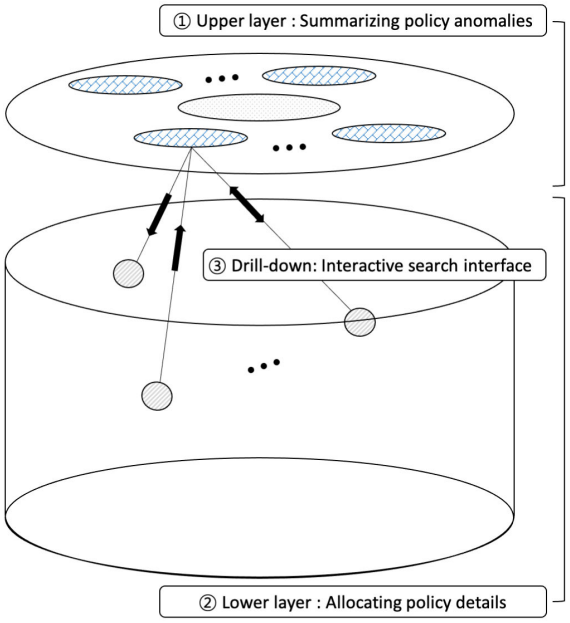


FIGURE 2. Conceptual image of hierarchical visualization methodology with drill-down interactive interface.

① Upper layer: represents the score summary of anomaly between the policies to give easy understanding of the overall policy status quickly. The layer consists of two components; i.e., the status summary dashboard and the anomaly bulletin board, as will be discussed in Section III-B.

② Lower layer: takes a detailed look at the property (e.g., usage and connectivity) of the policy which is mapped to the graphical element, and each policy is connected as a graph by the types of anomalies. The detail will be dealt with in Section III-C.

③ Drill-down interface: provides the interactive exploration of a complex policy anomaly to the administrator by examining the anomaly from a higher level of grouping (i.e., upper layer) to more detail level (i.e., lower layer); additionally, the interface supports a reverse drill-down as well, as will be described in Section III-D.

B. UPPER LAYER FOR COMPREHENSIVE ANOMALY VISUALIZATION

To understand the current status of the firewall policy immediately, it is important to grasp the overall anomaly between policies at once. For this, we design the upper layer consisting of two components, to provide the numerical summary of anomalies with its types based on the policy ID. Since we focus on the policy caused an anomaly in our method, the anomaly calculated by its definition [5] is sorted and subordinated by the priority of policy.

The status summary dashboard, as shown ① in Figure 3, displays the number of policies on the current ACL (i.e., *ACL Object*) and how many policies have anomalies

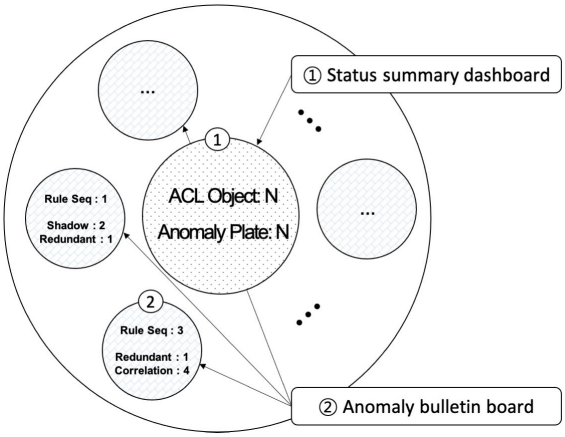


FIGURE 3. Visual composition of upper layer components.

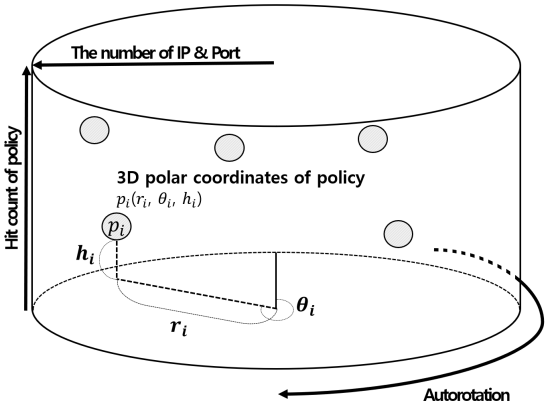


FIGURE 4. Policy allocation method on 3D lower layer.

(i.e., *Anomaly Plate*). The board specifically posts the ACL information as a numerical value for easy understanding of the overall policy status. From this information, the administrator of firewall can grasp how many policies suffer from anomalies and start trying to check the details of each policy. The anomaly plate specifically defines the number of the anomaly bulletin board (as shown ② in Figure 3), because the anomaly depends on each policy.

The anomaly bulletin board is made up of the policy and its anomaly information. The *Policy seq* stands for the priority (i.e., ID) of policy, and the types of anomalies (e.g., shadowing, redundancy, correlation, and generalization) with its number of occurrences belonged to the policy are notified at the bottom of the bulletin board; for instance, the bulletin board marked as ② indicates that the second policy is intertwined as one redundancy and four correlation anomalies with other policies. In other words, each bulletin board represents a single policy, and it is generated only when the policy involves anomaly; in worst case, the bulletin board is able to be generated up to the number of policies. One must be aware though, that the visualization controls the size of the bulletin boards flexibly considering the number of anomalies, in order to avoid an overlapping of the boards for better visibility. In addition, the bulletin board functions as a root for drill-down search by linking up policies sharing an anomaly.

C. POLICY ALLOCATION ON 3D LOWER LAYER

From the abstraction of the policy with anomaly status, every policy which share an anomaly are linked together through the corresponding bulletin board. After that, policies are visualized on the lower layer as 3D objects. Our approach, especially, tries to allocate policies to 3D space in order to provide crucial information from simply its location; thus relatively significant policies based on its usage and range, as shown in the Figure 4.

Meanwhile, there are generally three requirements for an intuitive visualization [42]–[44], such that (1) information in the text should not be omitted, (2) must be more intuitive than text, and (3) minimize the overlapping or screening of the screen to avoid distortion or loss of information. However, it is hard to visualize the ACL with satisfying basic three requirements, because there are a lot of policies and too complex in firewalls of large-scale networks.

To overcome the problem, every policies are placed following the Algorithm 1, not only to avoid an overlapping, but also to support intuitive understanding for the administrator. We assumed that there are the ACL P, including n policies: $P = \{p_0, p_1, \ldots, p_{n-1}, p_n\}, \quad (p_i \,|\, p_i \in P)$. To build the 3D cylinder shape of the lower layer, we calculate the maximum value of *hit count* and *sum of the number of IP and port* from

Algorithm 1: AllocatePolicy

Inputs : An access control list P, where policies $P = \{p_0, p_1, \cdots, p_n\}$, User-defined variable δ.
Output: The lower layer C included all coordinates of P.

```
begin
    Initialization of Cylinder C;
    r_max ⟵ MAX(CountIP/Port(P));
                                   /* radius */
    h_max ⟵ MAX(HitCount(P));
                                   /* height */
    C ⟵ MakeCylinder(r_max, h_max);
                        /* generate lower layer */
    foreach p_i in the set P do
        r_i ← SetRadius(p_i, r_max, δ);        ▷ eq. 7
        h_i ← SetHeight(p_i, h_max, δ);        ▷ eq. 8
        θ_i ← SetTheta(i, Size(P));            ▷ eq. 9
        AllocatePolicy(C, r_i, h_i, θ_i)
        ;/* allocate policy into the 3D
         lower layer                           */
    return C;
```

P as shown in *initialization of cylinder* of Algorithm 1. After that, we determine the 3D coordinates of the policy location based on a characteristic of the policy. Particularly, the radius value of the policy is calculated by the range of the policy as following the Equation 7:

$$r_i = \frac{\#\ of\ IP/Port\ in\ p_i}{MAX(\#\ of\ IP/Port\ in\ P)} \times \delta \tag{7}$$

where $MAX(\#\ of\ IP/Port\ in\ P)$ is the maximum value of IP and port ranges in the ACL, $\#\ of\ IP/Port\ in\ p_i$ is the range value of IP and port of the policy, and δ is the user-defined variable to adjust the proportion of the cylinder. For example, if the policy covers a wide range of bandwidth, then it will be located outside of the cylinder. Moreover, The method (i.e., Equation 8) for determining the height value of the policy is as follows:

$$h_i = \frac{p_i\ hit\ count}{MAX(P\ hit\ count)} \times \delta \tag{8}$$

where $MAX(P\ \ hit\ \ count)$ is the maximum value of the total usage for the ACL, $p_i\ \ hit\ \ count$ is the usage value of the policy, and δ is the user-defined variable the same as Equation 7. In this case, the more used the policy is, the higher the policy is located on. From the visualization method, we can easily discriminate policies based on the location of the policy (object). Furthermore, it is important to prevent an overlapping between objects, because there is a change of allocating policies to the same coordinates, even three-dimension. Hence, we define the angle θ of the policy from the starting point (the 12 o'clock position, clockwise) in the cylinder as following the Equation 9:

$$\theta_i = \frac{i}{n+1} \times 2\pi \tag{9}$$

where n is the total number of policies in the ACL and i is an order (i.e., priority) of the policy. Since there is no same priority policy in the ACL, it is impossible to overlap each other. Finally, all policies in the ACL are displayed in the coordinates determined by equations and the procedure of the Algorithm 1 and also they are connected to the bulletin board for representing the relationship between policies and anomalies intuitively. This can be utilized as a bridge, to support the optimization of the ACL, followed by the drill-down exploration we will discuss in the next section.

D. DRILL-DOWN USER INTERFACE

The most time-consuming task of optimizing the ACL is to check all policies one by one in ACL. This means that the administrator needs to find an anomaly in every row (line) on the text-based tabular user interface. Assuming there are m policies, in order to optimize policies, the administrator needs $\sum_{k=1}^{m}(k-1)$ comparisons and the time complexity of comparisons is $\mathcal{O}(n^2)$ in average case; for example, let's suppose that there is the ACL including 50 policies, then about 2,500 times of comparisons are required. Moreover, since some fields (e.g., IP and port) consist of the range value, such as 192.168.1.*, the number of comparisons could be larger. Even though he/she can discover an anomaly between policies, they should spend time again to revise them in order to prevent a security vulnerability.

In our method, however, most of the time-consuming task is eliminated. Since the hierarchical visualization provides anomaly information with related policies on the upper layer (summary) and the lower layer (detail), the administrator only needs to check and revise problems following drill-down searching. Specifically, the administrator firstly checks the dashboard in the upper layer, then he/she can confirm that how many and what kind of policies and anomalies are in the ACL at a glance. Next, he/she select (technically click) a bulletin board which is desired to revise, hereafter he/she can immediately identify the detail information, for instance, which policies are connected as an anomaly, what is the specification of the policy (usage or range) and so on. Finally, the administrator can modify or delete policy to optimize the whole ACL based on the above information. Thus, using the drill-down approach, they can manage the ACL, even large, with only a few steps (two or three).

Furthermore, the method is able to support the reverse drill down (same as drill up) method to solve an anomaly on the ACL, as shown in the Figure 2. First, for instance, administrators visualize all of the polices on the lower layer. After that, they can discriminate an important role (or desired) of policy in the lower layer based on the height and the radius. At the time of they choose the policy, a tree structure (i.e., linked graph) is formulated between the bulletin board (anomaly information) and every related policy. The administrator lastly can try the optimization task with the key information from the visualization.

Combining with the hierarchical visualization, the drill-down user interface can be used effectively to manage the ACL without a heavy time-consumption and high labor intensity. The administrator is readily accessible to a critical anomaly in the policy list and understands comprehensive information for optimizing the list. Since we postulate a real-world application, this practical method will be implemented considering in a large-scale network and evaluated to verify the performance in the use of ACL optimization.

IV. REAL-WORLD SCENARIOS WITH IMPLEMENTED SYSTEM

To verify a real-world feasibility, we introduce the system implementation based on the proposed method, and then show a case study of optimizing a complex real-world example using the implemented system.

A. SYSTEM ARCHITECTURE

The implemented system, namely F/Wvis, consists of four components: *log collector*, *data preprocessor*, *data analyzer*, and *data visualizer* as illustrated in Figure 5. The components are connected together with a common database, to process both the realtime network packets and complex policy. The log collector gathers two kinds of data from the firewall, such

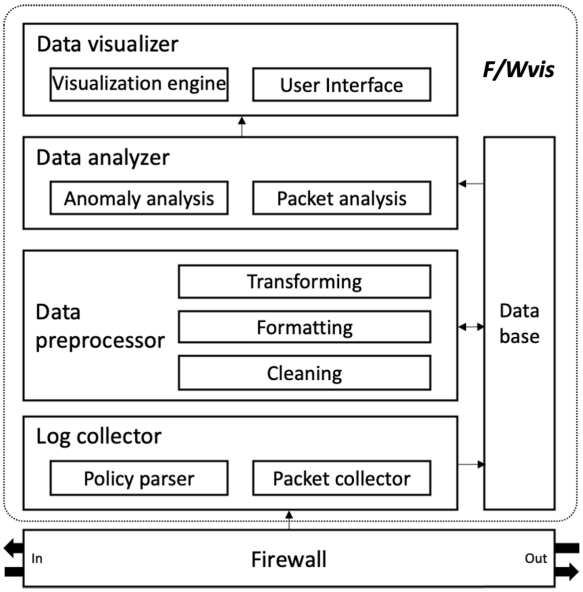


FIGURE 5. System architecture diagram.

as an ACL (syslog) and a packet (network traffic) passing through the firewall. After that, each data is processed to remove noise, unify a format from different sources, and reorganize a field structure; after which the processed data is stored in the database. The data analyzer tries to calculate not only the four types of the policy anomalies based on the ACL, but also hit counts and the number of used IP or port on the real-time network packets. Finally, the data visualizer builds the interactive user interface by applying the 3D visualization of the upper, lower layers with the anomaly information.

To improve the F/Wvis usability in practice, we consider minimum hardware and software requirements (e.g., Intel i7 core, 32GB memory, and so on). In particular, we try to compose open source software (Maria DB[1], Redis[2], Tomcat[3], and so on) for easy administration and maintenance. Moreover, Unity[4], which is the 3D game engine supporting low-end specification hardware, is employed to implement the 3D visualization and support seamless user interaction.

B. CASE STUDY WITH DRILL-DOWN EXPLORATION

We introduce how to optimize the policy anomalies using the F/Wvis, in order to solve the realistic problems (i.e., $\Re$ 1, 2, 3, and 4) presented in Section III. For the case study, we generated synthetic data (ACL) which includes 98 policies containing different types of 7 anomalies, such as shadowing, redundancy, generalization. As shown in Figure 6, the layout of the F/Wvis consists of primarily six components, including searching and filtering tools (Components A, B, and C), hierarchical drill-down interface (Components D and E; D is upper layer that has anomaly plates; E is lower layer that has ball and cube shaped IP objects), and windows for displaying policy and anomaly details (Component F) with visual options (Component G).

For the optimization of ACL, the administrator initiates their ACL to the F/Wvis system. The system momentarily visualizes the policies and anomalies on the layout (user interface), and hereat he/she almost immediately identifies how many policies are on the ACL and which policy has anomalies with other policies from the components B or D (solving $\Re$ 1). Next, they can choose the way how to start the optimization process; for instance, they try to search a target policy (or bandwidth) by submitting an IP address or port number to the searching tool (Component A), otherwise, they click the anomaly bulletin board on the upper layer (component D) to figure out the detail information of anomalies (disposing $\Re$ 2).

In the Figure 6, the administrator clicked the 77th policy to optimize three kinds of anomalies as displayed on the bulletin board, and at this moment he/she intuitively understands that the 77th policy has an anomaly connection with a high usage and a wide range (IP or port) policy from component E. Because the connected policies are located at the top and outer part of the lower layer. At the same time as the click event by them, the system opens the *X-ray view* as a pane (Component F) to provide the details, such as which policies are connected as an anomaly, what kind of anomalies are occurred, where the policy covers (e.g., IP or port range), and so on (supporting $\Re$ 4). Moreover, the details of anomalies are divided into its types as shown in component F, therefore the administrators perform the policy optimization (i.e., adding, removing and modifying) by clicking each anomaly type on X-ray view (mitigating $\Re$ 3); additionally, in order to boost the visibility, they can change the colours of policy on component D and E by its in/out traffic property using component G.

FIGURE 6. The layout of the F/Wvis and key components.

V. COMPREHENSIVE EVALUATION

A. PRELIMINARIES

1) PARTICIPANTS

For sophisticated evaluation, we recruited six participants - security experts from six different organizations. The purpose of the research was fully explained to the

[1]MariaDB, https://mariadb.org, accessed July 22, 2021.
[2]Redis, https://redis.io, accessed July 22, 2021.
[3]Tomcat, https://tomcat.apache.org, accessed July 22, 2021.
[4]Unity core platform, https://unity.com, Accessed July 22, 2021.

participants, after which we got their consent to use the survey and test results for our research without any honorarium. Since all of the participants are currently working for network (security) administrators, they are used to dealing with the policy of the firewall.

They have an average of 9.3 years of work experience ($\sigma = 3.9$) for network administrators with managing security devices, such as firewalls (including policy management), Intrusion Detection System (IDS), and anti-Distributed Denial of Service (Anti-DDos) system. All participants had managed at least a class C network (even multiple class B networks), operated by Windows and Unix-like systems. On average there are 171 policies ($min. = 64, max. = 335$) established and managed by the six experts.

2) DATA SET PREPARATION

We gathered six different data sets, denoted by O_1, O_2, O_3, O_4, O_5 and O_6, which are the access control lists (ACLs) operated in the field from the six organizations. They have a large-scale network containing terminals at least under two IPv4 class C network (i.e., 508 possible terminals), and the number of policies on the data sets (i.e., O_1, O_2, O_3, O_4, O_5) are 64, 88, 137, 148, 253, and 334 respectively. For security reasons arising from the nature of ACL, some fields are pseudonymized to prevent security threats; for example, a public IP address 210.119.29.* is converted to a private IP address like as 192.168.29.*, being careful not to lose their consistency.

The data sets are reorganized for evaluations with different facets of effectiveness, as two different kinds of purposes (Section V-C): how long it takes to discover an anomaly depending on (1) the ACL size (the number of policies) and (2) its types (shadowing, redundancy, correlation, and generalization). Based on the data set O_3, we produced five different sizes of data sets for the test (1). In detail, we prepared the five tasks with the five separate data sets, including one anomaly within 10, 25, 50, 70, and 100 policies, respectively. For the test (2), we created four types of data sets for finding different kinds of anomalies in the same number of policy; thus, each data set is formulated with 50 policies involved different types of anomalies

To validate usability of the F/Wvis for practical use (Section V-D), we provide the six data sets (O_1, O_2, O_3, O_4, O_5 and O_6) to the experts, and politely ask them to optimize the data sets using the F/Wvis system. In this evaluation, the data sets are served without any modification from the original (but pseudonymized) data sets.

B. SURVEY FOR QUALITATIVE ANALYSIS

The objectives of the survey are to evaluate the usability of the system and the user-friendliness of the interface. We prepared the questionnaire consisted of 4 questions for two categories, which are the system functions and the user interface. The question covers key issues regarding the practical use, such as performance of functions (e.g., deducing anomalies quickly), the user interface configurations (e.g., visualizing information intuitively), applicability to the real-world environment (e.g., large-scale networks), and so on.

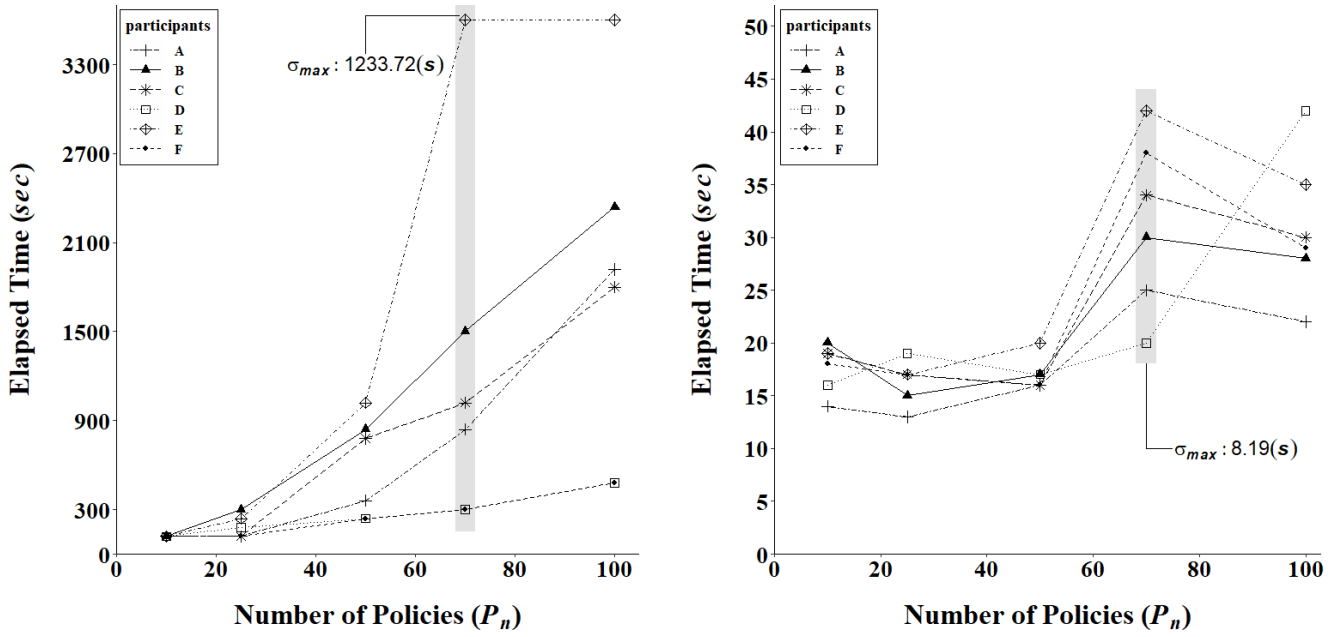
For this survey, a full explanation was given to the participants (i.e., six experts) about the fundamental concepts of the F/Wvis and its functions by face-to-face. Moreover, they learned how to use the F/Wvis for the optimization of firewall policy, and rehearsed all instructions including tasks for revising anomalies. We allow ample time (lasted 2.5 hours on average) to become skilled at the F/Wvis operation for not only this survey, but also later evaluations and tests. Each question is evaluated and quantitatively scored by five-point Likert scale [45], as 1 (very unuseful), 2 (unuseful), 3 (neither useful nor unuseful), 4 (useful), and 5 (very useful) respectively. We also asked them to give a short comment about the pros and cons, to take into account the details of user's assessments.

Table 2 shows the survey results, including quantitative scores and qualitative comments. On average, the F/Wvis earned relatively high scores in both categories, namely 4.29 and 4.5 for the function and the interface, respectively. In many parts of the question, we got the highest score (i.e., very useful); although there are a few lower scores (the lowest score is 3, only six times). Despite the variability in scores of some questions, the results are remarkable; particularly in terms of the user interface. According to the comments from the respondents *A*, *B*, and *D*, they were satisfied with the functions of calculating and providing anomalies with the drill-down methodology. Furthermore, the experts *E* and *F* mentioned that the F/Wvis could be taken advantage of to support a real-world complex network (or policies).

TABLE 2. Survey results of qualitative evaluation about system function and interface. (the questionnaire is shown in the Appendix A).

Experts	Eval. of function		Eval. of Interface		Comments (pros and cons)
	0 (very unuseful) to 5 (very useful)				
	Question	Score	Question	Score	
A	Q.1	5	Q.1	5	*easy to understand anomalies between policies*
	Q.2	5	Q.2	5	
	Q.3	5	Q.3	4	
	Q.4	4	Q.4	5	
B	Q.1	4	Q.1	4	*simple to check detail information based on drill-down interface*
	Q.2	5	Q.2	5	
	Q.3	5	Q.3	5	
	Q.4	4	Q.4	4	
C	Q.1	5	Q.1	5	*seems convenient, but not sure to apply real-world environment*
	Q.2	5	Q.2	5	
	Q.3	4	Q.3	5	
	Q.4	4	Q.4	3	
D	Q.1	4	Q.1	4	*good for finding unrevealed anomalies using search function*
	Q.2	4	Q.2	5	
	Q.3	4	Q.3	3	
	Q.4	3	Q.4	5	
E	Q.1	4	Q.1	5	*possible to analyze deep-level complex policies*
	Q.2	3	Q.2	5	
	Q.3	4	Q.3	5	
	Q.4	5	Q.4	4	
F	Q.1	5	Q.1	4	*will be proper to large-scale networks than small-scale*
	Q.2	4	Q.2	5	
	Q.3	5	Q.3	3	
	Q.4	3	Q.4	5	
Avg.	4.29		4.5		

On the other hand, there were negative comments about some questions as well. In particular, we got a low point (i.e., 3) twice to the issues about the real-world usability



(a) Optimization result by the popular system (Paloalto).

(b) Optimization result by the F/Wvis.

FIGURE 7. Elapsed time of optimization policy anomalies depending on different sizes of ACLs tested by six experts. (the actual results with specific numbers are shown in the Appendix B).

(involved in Q.4 of the evaluation of function). The concern from the *C* is worthy to be considered to improve the system utilizing in practice, regardless of the high quantitative scores. He added that although the F/Wvis is irrefutably useful to manage the firewall policy, the 3D graphical display could be unaccustomed to the management of policy immediately; especially for the administrator who worked with the text-based system for a long time.

The survey results gave us a much-needed variety of perspectives for the proposed method, including both positive and negative opinions. However, the overall upbeat scores and comments encouraged us and offered a viable alternative to the managing and optimizing policy anomaly. Inspired by the positive survey results, we will carry out the direct performance comparisons between the popular system and the F/Wvis in the next section.

C. COMPARATIVE TEST

To evaluate the performance of the F/Wvis in real-world usage, we carried out the comparative test in terms of (1) the size of ACL (i.e., the number of policies) and (2) the anomaly types (i.e., four types). In evaluating our system, it is not important the speed of data loading because our system does not require real-time data processing. And also, Our system can already load more than 2000 rules per second, which is enough for practical use; so we omitted it. We measured the time it takes them to solve the anomaly by using each popular system and the F/Wvis. The Paloalto [35] system was chosen as a popular system for the test, because the majority of the participants have experience operating the system and there are no differences between (text-based) popular systems. We put a time limit on each test, but it provided plenty of time for the test. Note that, the optimizations of ACLs were performed differently by the anomaly type. As the test metrics, all anomalies *should be made explicit* correctly, the shadowing and redundancy anomalies *should be revised not to generate other anomalies*, and the correlation and generalization *have to be recorded*.

1) EFFECTS OF ACL SIZES

The participants tried to optimize the five different test ACLs, consisting of one anomaly within 10, 25, 50, 70, and 100 policies; specifically, one shadowing in 10 and 50 policies, one redundancy in 25 policies, one shadowing and one generalization in 70 policies, and one redundancy and one correlation in 100 policies. We measured the time they solve the anomaly with the time limit of 60 minutes for each ACLs.

The test results are shown in Figure 7 as two graphs, such as the result by the Paloalto (Figure 7(a)) and the F/Wvis (Figure 7(b)). The x- and y-axes of graphs indicate the number of policies and the elapsed time of optimizing anomalies, respectively. The distinguished lines represent the result of five participants. Moreover, we can easily identify the scale differences of the y-axis between them, according to the elapsed time of each test.

The Figure 7(a), in case of using Paloalto system, indicates that the participants were struggling to find an anomaly in the ACLs. First, they readily handled the small sizes of ACLs (i.e., 10 and 25 policies) in a brief space of time. All six discovered and modified an anomaly within 120 seconds, and

they completed 25 policies in 300 seconds. However, they spent a lot of time drastically when they tried to optimize the lager ACLs, including 50 and 75 policies. Even though two experts settled the anomaly with the 50 policies until 240 seconds, but it took 580 seconds (almost 10 minutes) to fix the anomaly on average. For the 75 policies, they consumed an average of 790 seconds even except the elapsed time of one participant who failed in time (i.e., 60 minutes). The 100 policies obviously demanded a lot more time than previous tests about 1,400 seconds, also except one failed person. One interesting issue was that the optimization of ACL using the popular system was deeply influenced by the proficiencies of the administrators. For example, the result from 75 policies showed a large gap between the participants almost 20 minutes (σ_{max}). In fact, two volunteers who had worked more than 15 years (D and F in Figure 7(a)) solved the anomaly in 100 policies within 480 seconds; whereas the participant who has less than four years career (E in Figure 7(a)) spent all his time from the test with 75 policies.

Using the F/Wvis, on the other hand, the result showed the elapsed time was significantly reduced than the popular system. Since the F/Wvis visualized the anomalies on the 3D interface directly, there was no reason to take time for discovering the anomaly in the ACLs. Regardless of the number of policies, all participants achieved the optimizing of all ACLs within 50 seconds. Particularly, they spent only about 17 seconds to solve 10, 25, and 50 policies, and then took about 30 seconds for the 75 and 100 policies. The longest time to process the anomaly using the F/Wvis was 42 seconds by the person who spent one hour in the previous test (i.e., E in Figure 7(a) and 7(b)). This was nearly 100-fold faster (thus, 42 and 3600 seconds) than the case of using the text-based system. From the remarkable results, we definitely identified the proposed system could make it possible to reduce time-consuming administrative overhead for optimizing ACLs. Moreover, the standard deviation of elapsed time by the F/Wvis was only 8.19 seconds in the case of 75 policies. This implies that a security vulnerability is able to be prevented even the administrator is not an advanced level to manage the firewall policies.

2) DIFFICULTY OF ANOMALY TYPES

With the noteworthy results of the previous test, we tried to verify the performance of the F/Wvis depending on the types of anomalies. This test was established with 50 policies, because some participants had difficulty in solving the anomaly with more 50 policies from the former test. The time limit was set up as 30 minutes also considering the result from the previous test. We inserted one for each kind of anomaly to the ACL (i.e., 50 policies), then gave the ACLs to the participants for measuring an elapsed time of the optimization.

The Figure 8 showed the results from both the Paloalto and F/Wvis. Not unlike the previous test, there was a huge difference in performance. For the test of the shadowing anomaly by the F/Wvis, the mean time of the six persons was 12.5 seconds including the processes, such as recognizing

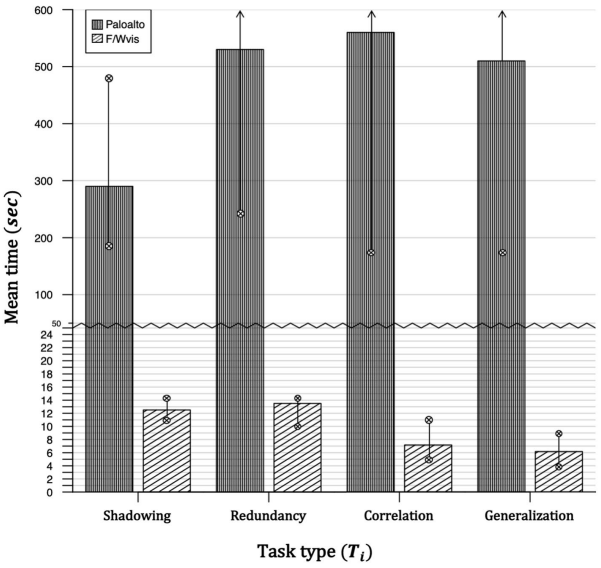


FIGURE 8. Test result of optimizing different types of anomalies. (the actual results with specific numbers are shown in the Appendix B).

the shadowing anomaly and revising it without any additional anomalies. The redundancy anomaly was treated until 13.5 seconds likewise the shadowing case. However, the participants spent about 290 and 530 seconds to optimize the anomalies of shadowing and redundancy respectively by the popular system. In the cases of correlation and generalization anomalies, the gaps of mean time were widened; for instance, the experts completed the finding and recording these anomalies to the note within 8 seconds by the F/Wvis. However, they figured out them consuming more than 500 seconds by the text-based system. From this test, we can identify that the shadowing and redundancy anomalies required to spend more time than generalization and correlation; because the participants needed to consider further impacts (anomalies) when a policy is modified or removed in the policy list. In addition, the difference by the proficiency level still appeared to the results from the popular system; such that, the standard deviation values were 116, 338, 323, and 285 seconds for each test in order at Figure 8. In contrast, it took only less than 3 seconds for all four tests by using the F/Wvis.

D. REAL-WORLD EFFECTIVENESS AND USABILITY

From the results of the survey and tests comparing with the text-based tool, we had confidence that the F/Wvis could be useful for the management of the firewall policy, and also utilized as a user-friendly tool for preventing a security vulnerability by optimizing anomalies on the ACLs.

To identify the assumptions, we performed an additional test that optimizes the firewall policies used in the real-world firewall. As we mentioned before, the six different ACLs were gathered from the six separated institutions for research purposes. The participants endeavored to discover anomalies on the different sizes of ACLs and fix them using both the

popular system and the F/Wvis. As same as the previous tests, they revised the shadowing and redundancy anomalies and noted the correlation and generalization cases, but up to 2 hours. The details of the data set they solved are shown in Table 3, including the number of policies and anomalies categorized by its types.

TABLE 3. The details of data sets from the real-world firewalls.

Data set	Number of policies	Number of anomalies			
		Shadowing	Redundancy	Correlation	Generalization
O_1	64	1	15	41	18
O_2	88	0	46	56	46
O_3	137	8	54	70	15
O_4	148	0	29	50	59
O_5	253	4	153	56	46
O_6	334	5	87	181	72

In the previous test, we discovered that using the text-based tool is a practical impossibility to revise even one anomaly in the ACL, which has more than 80 policies. Unsurprisingly, every participant in this test failed to optimize all the ACLs without the smallest one (i.e., O_1); nonetheless, they spent almost 2 hours on average to find anomalies in O_1. In addition, there was a postscript from the participant *D*: *'it seems near impossible to optimize the large-scale ACL involved more than 150 policies, even though we have enough time'*. From the result and comments from participants, we realized it is very hard to manage a ACL optimally based on text-based tools.

On the other hand, using the F/Wvis, all the participants succeeded to deal with every test ACLs within tens of seconds, even in the worst case (i.e., O_6). The results with the F/Wvis were plotted as a box-and-whisker graph as shown in Figure 9. The graphs indicated the mean time it takes for optimizing ACLs by six participants. We replaced the label of the x-axis to the number of policies in the test ACLs, to identify the effects of the ACL sizes easily. In most cases (e.g., O_1, O_2, O_3, and O_4) they optimized the anomalies in ten seconds. Even though the O_5 required to spend the most time, but it was only 35 seconds on average for six persons. An interesting point here, the number of policies is not simply affecting the time for optimizing the ACL. Although the size of O_6 is larger than O_5, it took less time than the smaller one (i.e., O_5); because O_5 has more anomalies than O_6, for example, 4 and 153 anomalies in O_5 (shadowing and redundancy, respectively) but only 5 and 87 anomalies in O_6. As a result, in using F/Wvis, both the number of anomalies and policies can influence on the optimizing ACLs, not only the size of ACL.

To verify the effects of the number of policies and anomalies on the elapsed time, we calculated the correlation between every two factors. As shown in the Figure 10, we can find two graphs about correlation analysis between the number of policies and elapsed time, and the number of anomalies and elapsed time, respectively. The correlation between policies and time in Figure 10(a) showed that the linear graph with the Pearson correlation coefficient value as

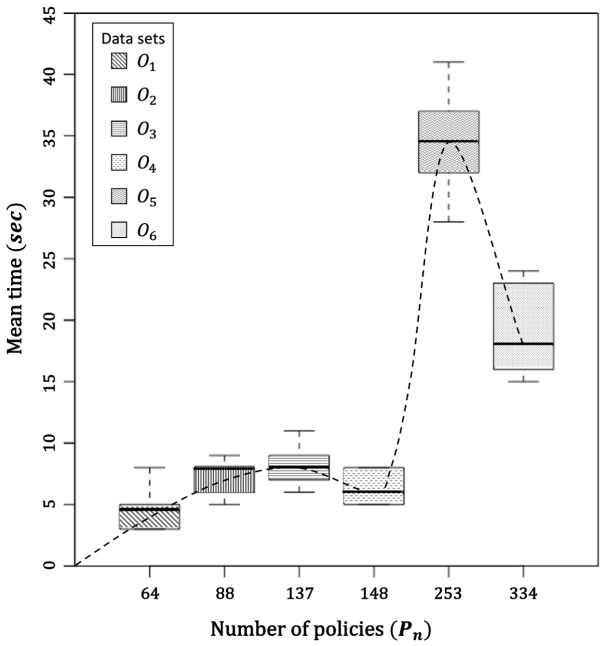


FIGURE 9. Test result of optimizing real-world firewall ACLs. (the actual results with specific numbers are shown in the Appendix B).

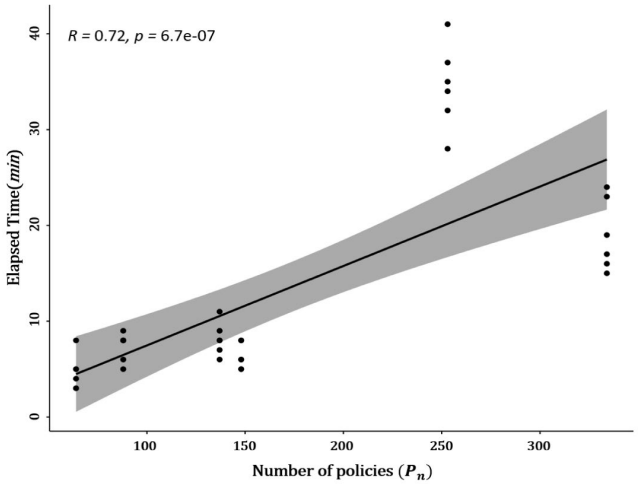
0.72. Meanwhile, we can find a strong positive relationship with 0.95 coefficient value as shown in Figure 10(b), for between anomalies and time. It is sure that the number of policies and anomalies are both important factors affecting the time consumption for optimizing the ACL using the F/Wvis; however, the number of anomalies has a great influence on the time consumption, rather than the number of policies.

VI. DISCUSSION

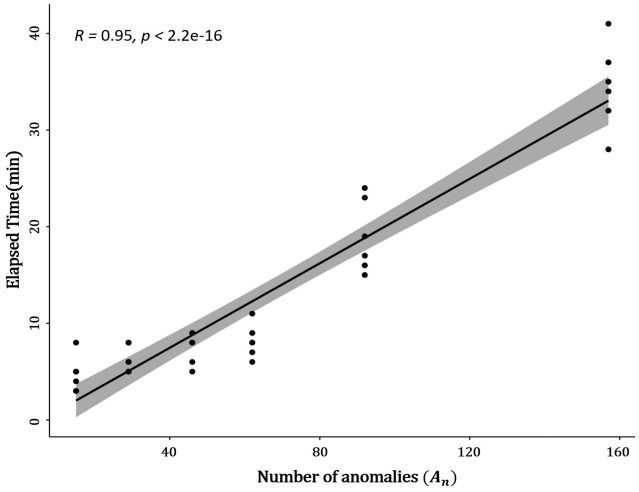
A. EVALUATION RESULT AND ANALYSIS

The survey and three different tests were thoroughly designed to identify the effectiveness and usability of the proposed method, considering real-world situations. In particular, the six experts were invited to evaluate the performance of the F/Wvis comparing the text-based firewall management tools (e.g., Paloalto); in terms of both qualitative and quantitative. After they answered the questionnaire for the F/Wvis, they performed the optimization of the synthetic ACLs using the popular tools and the F/Wvis respectively; at this time, we measured the elapsed time of the optimization as a performance of each tool. Moreover, we provided the real-world ACLs to estimate the usability of the F/Wvis, with the same test of the synthetic data. In the evaluations, we also endeavored to identify that what makes it difficult to optimize the ACL from different factors, such as the number of policies, the number of anomalies, and the types of anomalies. To verify this, we organized the synthetic data based on the different number of policies and anomaly types; furthermore, the six types of real-world ACLs were involved with the different numbers of anomalies and various types.

As a result, the F/Wvis showed a remarkable performance to optimize various ACLs, regardless of the ACL sizes,



(a) Correlation plot between the number of polices and elapsed time.



(b) Correlation plot between the number of anomalies and elapsed time.

FIGURE 10. Correlation analysis (Pearson correlation test) with confidence interval set at 95% (shadowed area) between policy, anomaly and elapsed time.

anomaly types, and the number of anomalies. In fact, the optimization time was significantly reduced than the popular tool; in spite of the optimization from the popular tool exceeded the time limit in some hard cases. The optimization by F/Wvis usually spent tens of seconds, however, the tasks by the text-based tool took several tens of minutes or even more than an hour. For instance, the worst cases of the tests (i.e., meantime except for time out cases) were such that 1,770 versus 29.16 seconds (test 1), and 559.8 versus 7.16 seconds (test 2) from the text-based tool and the F/Wvis respectively. In real-world problems, especially, the F/Wvis finished all types of data sets within 30 minutes; on the other hand, the text-based tool was failed all tests, except the smallest one.

From these results, we not only proved the correlation between the optimization time and the number of policies, and the optimization time and the number of anomalies but also showed that these correlations can be reduced through F/Wvis. In addition, it can also reduce the optimization time caused by differences in the proficiency of administrators, which proves that it is an effective decision-making tool in various practice environments. All detailed results of the test are attached in Appendix B.

B. FURTHER CONSIDERATIONS

1) VISUALIZATION FOR FIREWALL MANAGEMENT

For the easy and effective management of the firewall policy, various studies and applications had developed based on the state-of-the-art techniques of the time. The early researches tried to analyze an anomaly in the policy list and provided as textual information by a command line or table format [8], [37]. With the development of user interface methods, the management of the ACL had become easy with the interactive user interfaces and graphical visualization techniques [10], [11]. They actively utilized the visual components to express the firewall policy and anomaly using a different shape, colour, and its combination. Recently, the applications were combined with three-dimensional graphics to represent the various information in the firewall ACLs [9], [38]. From the visualization approaches, the administrator can understand the situation of firewall easily, and prevent a security vulnerability caused by firewall anomalies. Even though we proposed and identified the benefits of our three-dimension, interactive visualization approach, the visualization method is needed to be considered and improved to support a security administration of various, large-scale complex networks. We will further investigate and extend the methodology of firewall policy visualization for the future work.

2) RECOMMEND SYSTEM

The policy anomaly is already well-defined and used to prevent a security problem, however, it is difficult to manage them optimally. Since the policy could be defined with multivariate fields such as the band of IP address, multiple ports or protocols, it is really hard to optimize them at a glance. Moreover, the policy has an effect on other policies defined later; so it is needed to a priority of policy when we revise, modify, or remove a policy on the ACL. Hence, the recent studies and applications only support the indicating or visualizing an anomaly or policy to the firewall administrator [20], [46], [47]. Using the information, they need to optimize the policy list manually, even in the F/Wvis we proposed. Wherefore, now we are engaged to utilize the Artificial Intelligence (AI) and Machine Learning (ML) techniques, in order to discover not only an anomaly, but also additional effects on the ACL when we modified them. In particular, we assume the system that recommends a way to revise or remove a policy, to maintain the optimal policy list of the firewall.

3) REGARDING VERY LARGE NETWORK

Even though this study considered a large-scale network (such as more than 100 terminals), very large networks have

operated in the real-world environment. We employed a single firewall in this work, therefore there is a limitation to maintain the policies (about several hundreds). Nowadays, however, there are huge networks combining with multiple firewalls and other security devices [48], [49]. In this case, it is impossible to manage the big ACL using both the text-based tool and the F/Wvis also, because of the limitation of visual space on the user interface. To consider the real-world situation in near future, we need to develop a methodology to cover the security of a very large network; considering various techniques, such as visualization, recommendation, AI and ML, and others widely.

VII. CONCLUSION REMARKS

In this paper, we introduced a novel visual approach to optimize firewall policies. The hierarchical visualization methodology and drill-down user interface were proposed to support intuitive management of the firewall ACL. The evaluation and the user survey were designed to identify the usability and effectiveness of the proposed method in the real-world environment. In particular, the tests were organized to validate the performance of our method (i.e., the F/Wvis) in terms of three different aspects, such as the number of policies, anomalies, and anomaly types. From the evaluation, we reveal that the F/Wvis can solve an anomaly quickly and manage the ACL easily comparing with the text-based traditional tools. Moreover, the factors affecting the time-consuming for optimizing a policy were investigated based on the correlation test. As an additional benefit to use the F/Wvis, it does not require an advanced level of proficiency for the firewall administrators. From the positive reviews of the qualitative survey, we gained confidence that the F/Wvis can be utilized for the management of real-world firewall in practice.

APPENDIXES

APPENDIX A

SURVEY QUESTIONNAIRE

Instructions:

- We showed the demonstration video as a visualized material to introduce our system.
- We explained the detail of system functions and interfaces face-to-face
- We gave a enough time (about 30 minutes) to understand the operation of the F/Wvis
- We provided an example ACL with one anomaly (shadowing) and gave a time to solve the problem
- We asked them to answer the survey questionnaire and some comments.

Questionnaire: see Figure 11.

Technical survey of the application for firewall policy management

INSTRUCTION:

1. Please watch the visual guide (about 5 minute).
2. We will explain the interfaces and functions of tool (20 minutes).
3. Please learn and test the interface and functions of tool (30 minutes).
4. Please try to optimize the anomaly in the access policy (25 minutes).
5. Finally, please answer the following questions

EVALUATION METHOD:

- This survey is categorized as the function and interface of parts.
- We will ask four questions for each part
- You can give a point to each question between 0 (very useless) ~ 5 (very useful)
- We appreciate your additional comments about pros and cons of the tool

	Question	Score
Function	Q.1: Does the anomaly was provided accurately and consistently?	
	Q.2: Does the search function was useful to discover the policy based on IP or Port?	
	Q.3: Do the filters and drill-down functions were convenient?	
	Q.4: Is the tool proper to the management of real-world firewall?	
Comment		
User Interface	Q.1: How intuitive can you identify an anomaly based on the v visualization of tool?	
	Q.2: Is it easy to check the detail information of each policy?	
	Q.3: Is there enough information on the display to revise an anomaly and manage a policy list?	
	Q.4: Is there a user-friendly interface with comfortable interactions for the security administrators?	
Comment		

FIGURE 11. The questionnaire for surveying the F/Wvis function and interface.

APPENDIX B

COMPARATIVE TEST MATERIALS

A. TEST PAPER (ACL)

Test paper: see Table 4.

Instructions:

- The ACL as shown in the Table 4 was organized for the evaluation.
- All test papers (ACL) were extracted and organized based on the ACL (Table 4).
- Every test paper was uploaded to the Paloalto and the F/Wvis system for each test.
- The column *'Associated policy'* (solutions, bold) was not provided to the participants.
- The test paper is made by extracting policies from the six real-world data.
- Some fields are pseudonymized (changed) to prevent information leaks.

B. DETAILS OF TEST RESULTS

1) ACTUAL RESULT FOR VARIABLE SIZE OF ACL

Setting up test sheets:

- Optimize 10 access policies on the ACL: one shadowing anomaly problem
- Optimize 25 access policies on the ACL: one Redundancy anomaly problem
- Optimize 50 access policies on the ACL: one shadowing anomaly problem
- Optimize 75 access policies on the ACL: one shadowing and one generalization anomalies problem
- Optimize 100 access policies on the ACL: one redundancy and correlation anomalies problem

The detail results: see Table 5.

TABLE 4. The test paper consists of 100 policies with n anomalies.

ID	protocol	s_IP	s_port	d_IP	d_port	action	Associated policy
1	**TCP**	**1.1.1.41**	**ANY**	**2.2.2.8**	**22**	**ALLOW**	**Redundancy : 1-25**
2	TCP	1.1.1.12	ANY	2.2.2.34	80	DENY	
3	TCP	2.2.2.48	ANY	1.1.1.16	80	ALLOW	
4	TCP	2.2.2.74	ANY	1.1.1.15	139	ALLOW	
5	**TCP**	**2.2.2.48**	**ANY**	**1.1.1.17**	**22**	**ALLOW**	**Shadowing : 5-10**
6	TCP	1.1.1.13	ANY	2.2.2.9	143	ALLOW	
7	**TCP**	**1.1.9.40**	**ANY**	**2.2.2.75**	**143**	**ALLOW**	**Shadowing : 7-40**
8	TCP	1.1.1.12	ANY	2.2.2.34	8080	DENY	
9	TCP	1.1.1.172	ANY	2.2.2.188	139	DENY	
10	**TCP**	**2.2.2.48**	**ANY**	**1.1.1.17**	**22**	**DENY**	**Shadowing : 5-10**
11	TCP	2.2.2.148	ANY	1.1.1.198	22	ALLOW	
12	TCP	1.1.1.62	ANY	2.2.2.46	8080	ALLOW	
13	TCP	1.1.1.76	ANY	2.2.2.81	21	ALLOW	
14	TCP	2.2.2.6	ANY	1.1.1.19	21	DENY	
15	TCP	1.1.1.43	ANY	2.2.2.47	443	ALLOW	
16	TCP	1.1.1.101	ANY	2.2.2.10	143	ALLOW	
17	**TCP**	**1.1.1.86**	**ANY**	**2.2.2.50**	**8443**	**DENY**	**Generalization : 17-67**
18	TCP	1.1.1.86	ANY	2.2.2.91	443	ALLOW	
19	TCP	2.2.2.20	ANY	1.1.1.20	8443	ALLOW	
20	TCP	1.1.1.40	ANY	2.2.2.5	53	ALLOW	
21	TCP	1.1.1.64	ANY	2.2.2.98	21	ALLOW	
22	TCP	1.1.1.3	ANY	2.2.2.21	22	ALLOW	
23	TCP	2.2.2.36	ANY	1.1.1.87	23	ALLOW	
24	TCP	1.1.1.65	ANY	2.2.2.68	53	ALLOW	
25	**TCP**	**1.1.1.41**	**ANY**	**2.2.2.8**	**22**	**ALLOW**	**Redundancy : 1-25**
26	TCP	1.1.1.101	ANY	2.2.2.100	21	ALLOW	
27	TCP	1.1.1.37	ANY	2.2.2.11	8443	ALLOW	
28	TCP	1.1.1.37	ANY	2.2.2.82	80	ALLOW	
29	TCP	1.1.1.77	ANY	2.2.2.69	161	ALLOW	
30	TCP	1.1.1.244	ANY	2.2.2.48	22	ALLOW	
31	TCP	2.2.2.37	ANY	1.1.1.98	443	ALLOW	
32	TCP	1.1.1.40	ANY	2.2.2.54	8443	ALLOW	
33	**TCP**	**1.1.1.3**	**ANY**	**ANY**	**161**	**DENY**	**Correlation : 33-37**
34	TCP	1.1.1.57	ANY	2.2.2.93	21	ALLOW	
35	TCP	1.1.1.3	ANY	2.2.2.83	443	ALLOW	
36	TCP	2.2.2.24	ANY	1.1.1.61	161	ALLOW	
37	**TCP**	**ANY**	**ANY**	**1.1.1.3**	**161**	**ALLOW**	**Correlation : 33-37**
38	TCP	2.2.2.22	ANY	1.1.1.94	443	ALLOW	
39	TCP	1.1.1.101	ANY	2.2.2.34	80	ALLOW	
40	**TCP**	**1.1.9.40**	**ANY**	**2.2.2.75**	**143**	**DENY**	**Shadowing : 7-40**
41	TCP	1.1.1.100	ANY	2.2.2.53	515	ALLOW	
42	TCP	1.1.1.89	ANY	2.2.2.99	161	ALLOW	
43	TCP	1.1.1.59	ANY	1.1.1.93	43	ALLOW	
44	**TCP**	**1.1.1.3**	**ANY**	**2.2.2.17**	**443**	**ALLOW**	**Redundancy : 44-99**
45	TCP	1.1.1.8	ANY	2.2.2.4	45	ALLOW	
46	TCP	1.1.1.37	ANY	2.2.2.84	46	DENY	
47	TCP	1.1.1.40	ANY	2.2.2.12	161	ALLOW	
48	TCP	1.1.1.89	ANY	2.2.2.62	80	ALLOW	
49	TCP	1.1.1.40	ANY	2.2.2.75	80	ALLOW	
50	TCP	2.2.2.73	ANY	1.1.1.55	443	ALLOW	
51	TCP	1.1.1.68	ANY	2.2.2.95	25	ALLOW	
52	TCP	1.1.1.7	ANY	2.2.2.71	443	ALLOW	
53	TCP	1.1.1.81	ANY	2.2.2.25	53	ALLOW	
54	TCP	1.1.1.36	ANY	2.2.2.56	8443	ALLOW	
55	TCP	1.1.1.37	ANY	2.2.2.15	443	ALLOW	
56	TCP	1.1.1.69	ANY	2.2.2.13	8000	ALLOW	
57	TCP	2.2.2.74	ANY	1.1.1.15	22	ALLOW	
58	TCP	1.1.1.59	ANY	2.2.2.26	58	DENY	
59	TCP	1.1.1.6	ANY	2.2.2.57	443	ALLOW	
60	TCP	1.1.1.25	ANY	2.2.2.14	25	ALLOW	
61	TCP	1.1.1.89	ANY	2.2.2.76	25	ALLOW	
62	TCP	2.2.2.78	ANY	1.1.1.24	443	ALLOW	
63	TCP	2.2.2.88	ANY	1.1.1.35	8999	ALLOW	
64	TCP	1.1.1.82	ANY	2.2.2.40	64	ALLOW	
65	TCP	1.1.1.70	ANY	2.2.2.27	990	ALLOW	
66	TCP	1.1.1.37	ANY	2.2.2.58	21	ALLOW	
67	**TCP**	**1.1.1.86**	**ANY**	**2.2.2.50**	**8443**	**ALLOW**	**Generalization : 17-67**
68	TCP	1.1.1.92	ANY	2.2.2.16	68	ALLOW	
69	TCP	1.1.1.83	ANY	2.2.2.28	69	DENY	
70	TCP	1.1.1.46	ANY	2.2.2.41	443	ALLOW	
71	TCP	1.1.1.71	ANY	2.2.2.1	8999	ALLOW	
72	TCP	1.1.1.71	ANY	2.2.2.77	80	ALLOW	
73	TCP	2.2.2.60	ANY	1.1.1.48	8000	ALLOW	
74	TCP	1.1.1.34	ANY	2.2.2.29	161	ALLOW	
75	TCP	1.1.1.47	ANY	2.2.2.64	993	ALLOW	
76	TCP	1.1.1.91	ANY	2.2.2.85	995	ALLOW	
77	TCP	1.1.1.72	ANY	2.2.2.19	990	ALLOW	
78	TCP	1.1.2.0	ANY	2.2.2.19	990	ALLOW	
79	TCP	1.1.1.59	ANY	2.2.2.3	10022	ALLOW	
80	TCP	1.1.1.91	ANY	2.2.2.59	8999	ALLOW	
81	TCP	2.2.2.142	ANY	1.1.1.193	22	DENY	
82	TCP	2.2.2.45	ANY	1.1.1.99	10022	ALLOW	
83	TCP	1.1.1.84	ANY	2.2.2.43	443	ALLOW	
84	TCP	1.1.1.96	ANY	2.2.2.96	443	ALLOW	
85	TCP	1.1.1.28	ANY	2.2.2.18	21	ALLOW	
86	TCP	1.1.1.73	ANY	2.2.2.86	222	ALLOW	
87	TCP	1.1.1.103	ANY	2.2.2.55	161	ALLOW	
88	TCP	1.1.1.231	ANY	2.2.2.19	990	DENY	
89	TCP	2.2.2.33	ANY	1.1.1.39	161	DENY	
90	TCP	1.1.1.74	ANY	2.2.2.44	8443	ALLOW	
91	TCP	2.2.2.80	ANY	1.1.1.54	8888	DENY	
92	TCP	1.1.1.31	ANY	2.2.2.79	222	ALLOW	
93	TCP	1.1.1.29	ANY	2.2.2.32	443	ALLOW	
94	TCP	1.1.1.62	ANY	2.2.2.46	22	ALLOW	
95	TCP	2.2.2.61	ANY	1.1.1.53	161	ALLOW	
96	TCP	1.1.1.2	ANY	2.2.2.66	443	ALLOW	
97	TCP	1.1.1.55	ANY	2.2.2.166	161	DENY	
98	TCP	2.2.2.39	ANY	1.1.1.35	161	DENY	
99	**TCP**	**1.1.1.3**	**ANY**	**2.2.2.17**	**443**	**ALLOW**	**Redundancy : 44-99**
100	TCP	1.1.1.1	ANY	2.2.2.87	80	DENY	

TABLE 5. Test result for variable size of ACL.

Task (policies)	System	Elapsed time per participants						Statistics	
		A	B	C	D	E	F	μ (Mean)	σ (Std.)
10	Paloalto	2(m)	2(m)	2(m)	2(m)	2(m)	2(m)	1	0
	F/Wvis	14(s)	20(s)	19(s)	16(s)	19(s)	18(s)	17.66	2
25	Paloalto	2(m)	5(m)	2(m)	3(m)	4(m)	2(m)	3	1.26
	F/Wvis	13(s)	15(s)	17(s)	16(s)	17(s)	17(s)	15.83	1.5
50	Paloalto	6(m)	14(m)	13(m)	4(m)	17(m)	4(m)	9.66	5.68
	F/Wvis	16(s)	17(s)	16(s)	18(s)	20(s)	16(s)	17.16	1.5
75	Paloalto	14(m)	25(m)	17(m)	5(m)	Time out	5(m)	21	20.56
	F/Wvis	25(s)	30(s)	34(s)	38(s)	42(s)	38(s)	34.5	5.7
100	Paloalto	32(m)	39(m)	30(m)	8(m)	Time out	8(m)	29.5	19.75
	F/Wvis	22(s)	28(s)	30(s)	31(s)	35(s)	29(s)	29.16	3.9

*The time limit is one hour per each task.

2) ACTUAL RESULT FOR DIFFERENT TYPES OF ANOMALIES

Setting up test sheets:

- Task 1: Please find and revise one shadowing anomaly
- Task 2: Please find and revise one redundancy anomaly
- Task 3: Please find and revise one correlation anomaly
- Task 4: Please find and revise one generalization anomaly

Detail results: see Table 6.

(*All tasks were performed with 50 policies of ACLs)

TABLE 6. Test result for types of anomalies.

Task	System	Elapsed time per participants						Statistics	
		A	B	C	D	E	F	μ (Mean)	σ (Std.)
1	Paloalto	6(m)	4(m)	3(m)	5(m)	8(m)	3(m)	4.83	1.77
	F/Wvis	12(s)	12(s)	13(s)	11(s)	15(s)	13(s)	12.66	1.24
2	Paloalto	5(m)	4(m)	6(m)	17(m)	15(m)	6(m)	8.83	5.14
	F/Wvis	10(s)	15(s)	13(s)	14(s)	14(s)	15(s)	13.5	1.7
3	Paloalto	9(m)	7(m)	13(m)	6(m)	18(m)	3(m)	9.33	4.92
	F/Wvis	6(s)	5(s)	6(s)	6(s)	11(s)	9(s)	7.16	2.11
4	Paloalto	8(m)	7(m)	6(m)	10(m)	17(m)	3(m)	8.5	4.34
	F/Wvis	5(s)	4(s)	7(s)	5(s)	9(s)	7(s)	6.16	1.67

*The time limit is 30 minutes per each task.

3) REAL-WORLD USABILITY

Setting up test sheets:

- Task 1: Please optimize the real-world ACL from O_1
- Task 2: Please optimize the real-world ACL from O_2
- Task 3: Please optimize the real-world ACL from O_3
- Task 4: Please optimize the real-world ACL from O_4
- Task 5: Please optimize the real-world ACL from O_5
- Task 6: Please optimize the real-world ACL from O_6

Detail results: see Table 7.

TABLE 7. Test result for real-world usability.

Task	System	Elapsed time per participants (minutes)						Statistics	
		A	B	C	D	E	F	μ (Mean)	σ (Std.)
1	Paloalto	108	118	102	120	101	98	107.83	9.26
	F/Wvis	3	5	3	5	8	4	4.66	1.69
2	Paloalto	Time out						N/A	
	F/Wvis	8	8	8	5	9	6	7.33	1.37
3	Paloalto	Time out						N/A	
	F/Wvis	6	8	7	8	11	9	8.16	1.57
4	Paloalto	Time out						N/A	
	F/Wvis	5	6	5	6	8	8	6.33	1.24
5	Paloalto	Time out						N/A	
	F/Wvis	28	35	32	37	41	34	34.5	4.03
6	Paloalto	Time out						N/A	
	F/Wvis	15	19	16	17	24	23	19	3.4

*The time limit is 2 hours per each task.

REFERENCES

[1] M. Cooney. (Apr. 2020). *Network World 2020 State of the Network: SD-WAN, Edge Networking and Security are Hot*. Accessed: Jun. 24, 2020. [Online]. Available: https://www.networkworld.com/article/3537559/state-of-the-network-sd-wa n-edge-networking-and-security-issues-heat-things-up.html

[2] R. Hunt, "Internet/Intranet firewall security—Policy, architecture and transaction services," *Comput. Commun.*, vol. 21, no. 13, pp. 1107–1123, 1998.

[3] W. Stallings, *Network Security Essentials: Applications and Standards*, 4th ed. London, U.K.: Pearson, 2003.

[4] A. Voronkov, L. A. Martucci, and S. Lindskog, "Measuring the usability of firewall rule sets," *IEEE Access*, vol. 8, pp. 27106–27121, 2020.

[5] E. S. Al-Shaer and H. H. Hamed, "Firewall policy advisor for anomaly discovery and rule editing," in *Proc. Int. Symp. Integr. Netw. Manage.* Boston, MA, USA: Springer, 2003, pp. 17–30.

[6] A. Wool, "A quantitative study of firewall configuration errors," *Computer*, vol. 37, no. 6, pp. 62–67, Jun. 2004.

[7] Algosec. (2016). *Firewall Management: 5 Challenges Every Company Must Address*. Accessed: Aug. 23, 2020. [Online]. Available: https://www.algosec.com/wp-content/uploads/2016/03/Firewall-Management-5-Challenges-Every-Company-Must-Address-WEB.pdf

[8] T. Tran, E. S. Al-Shaer, and R. Boutaba, "PolicyVis: Firewall security policy visualization and inspection," in *Proc. LISA*, vol. 7, 2007, pp. 1–16.

[9] A. K. Meena and N. Hubballi, "NViZ: An interactive visualization of network security systems logs," in *Proc. Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2020, pp. 685–687.

[10] F. Mansmann, T. Göbel, and W. Cheswick, "Visual analysis of complex firewall configurations," in *Proc. 9th Int. Symp. Vis. Cyber Secur. (VizSec)*, 2012, pp. 1–8.

[11] U.-H. Kim, J.-M. Kang, J.-S. Lee, H.-S. Kim, and S.-Y. Jung, "Practical firewall policy inspection using anomaly detection and its visualization," *Multimedia Tools Appl.*, vol. 71, no. 2, pp. 627–641, Jul. 2014.

[12] S. Suri and G. Varghese, "Packet filtering in high speed networks," in *Proc. 10th Annu. ACM-SIAM Symp. Discrete Algorithms*, 1999, pp. 969–970.

[13] D. B. Chapman, E. D. Zwicky, and D. Russell, *Building Internet Firewalls*. Sebastopol, CA, USA: O'Reilly Media, 1995.

[14] G. Van Rooij, "Real stateful TCP packet filtering in IP filter," in *Proc. 10th USENIX Secur. Symp.*, 2001.

[15] D. Hartmeier and A. Systor, "Design and performance of the OpenBSD stateful packet filter (pf)," in *Proc. USENIX Annu. Tech. Conf., FREENIX Track*, 2002, pp. 171–180.

[16] T. Gillis, *Securing the Borderless Network: Security for the Web 2.0 World*. London, U.K.: Pearson, 2010.

[17] E. Karaarslan, T. Tuglular, and H. Sengonca, "Enterprise wide web application security: An introduction," in *Proc. 13th Annu. EICAR Conf.*, 2004, pp. 1–18.

[18] R. Sekar, Y. Guang, S. Verma, and T. Shanbhag, "A high-performance network intrusion detection system," in *Proc. 6th ACM Conf. Comput. Commun. Secur. (CCS)*, 1999, pp. 8–17.

[19] M. Stevens, "UTM: One-stop protection," *Netw. Secur.*, vol. 2006, no. 2, pp. 12–14, Feb. 2006.

[20] H. Hu, G.-J. Ahn, and K. Kulkarni, "Detecting and resolving firewall policy anomalies," *IEEE Trans. Dependable Secure Comput.*, vol. 9, no. 3, pp. 318–331, May/Jun. 2012.

[21] G. Goldszmidt and S. Jürgen, *Integrated Network Management VIII: Managing it All*, vol. 118. Boston, MA, USA: Springer, 2013.

[22] A. Wool, "Trends in firewall configuration errors: Measuring the holes in Swiss cheese," *IEEE Internet Comput.*, vol. 14, no. 4, pp. 58–65, Jul. 2010.

[23] A. Tongaonkar, N. Inamdar, and R. Sekar, "Inferring higher level policies from firewall rules," in *Proc. LISA*, vol. 7, 2007, pp. 1–10.

[24] E. S. Al-Shaer and H. H. Hamed, "Discovery of policy anomalies in distributed firewalls," in *Proc. IEEE INFOCOM*, vol. 4, 2004, pp. 2605–2616.

[25] E. S. Al-Shaer and H. H. Hamed, "Modeling and management of firewall policies," *IEEE Trans. Netw. Service Manage.*, vol. 1, no. 1, pp. 2–10, Apr. 2004.

[26] F. Valenza, S. Spinoso, and R. Sisto, "Formally specifying and checking policies and anomalies in service function chaining," *J. Netw. Comput. Appl.*, vol. 146, Nov. 2019, Art. no. 102419.

[27] D. Bringhenti, G. Marchetto, R. Sisto, F. Valenza, and J. Yusupov, "Automated optimal firewall orchestration and configuration in virtualized networks," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS)*, Apr. 2020, pp. 1–7.

[28] C. Togay, A. Kasif, C. Catal, and B. Tekinerdogan, "A firewall policy anomaly detection framework for reliable network security," *IEEE Trans. Rel.*, early access, Jul. 8, 2021, doi: 10.1109/TR.2021.3089511.

[29] E. Al-Shaer, H. Hamed, R. Boutaba, and M. Hasan, "Conflict classification and analysis of distributed firewall policies," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 10, pp. 2069–2084, Oct. 2005.

[30] L. Yuan, H. Chen, J. Mai, C.-N. Chuah, Z. Su, and P. Mohapatra, "FIREMAN: A toolkit for firewall modeling and analysis," in *Proc. IEEE Symp. Secur. Privacy (S&P)*, May 2006, p. 15.

[31] H. Hu, G.-J. Ahn, and K. Kulkarni, "FAME: A firewall anomaly management environment," in *Proc. 3rd ACM Workshop Assurable Usable Secur. Configuration*, 2010, pp. 17–26.

[32] F. Valenza, C. Basile, D. Canavese, and A. Lioy, "Classification and analysis of communication protection policy anomalies," *IEEE/ACM Trans. Netw.*, vol. 25, no. 5, pp. 2601–2614, Oct. 2017.

[33] E. Al-Shaer and H. Hamed, "Design and implementation of firewall policy advisor tools," DePaul CTI, Tech. Rep. CTI-TR-02-006, Aug. 2002.

[34] Check Point Software Technologies Ltd. (2015). *Firewall R77 Versions Administration Guide*. Accessed: Jun. 24, 2020. [Online]. Available: https://sc1.checkpoint.com/documents/R77/CP_R77_Firewall_WebAdmin/92703.htm

[35] Paloalto Networks. (2019). *Pa-Series User Manual*. Accessed: Jun. 24, 2020. [Online]. Available: https://www.paloaltonetworks.co.kr/network-security/pa-series

[36] AhnLab. (2019). *Trusguard Firewalls*. Accessed: Jun. 24, 2020. [Online]. Available: https://global.ahnlab.com/site/product/productSubDetail.do?prodSeq=5690

[37] W. Xu, M. Shehab, and G.-J. Ahn, "Visualization based policy analysis: Case study in selinux," in *Proc. 13th ACM Symp. Access Control Models Technol.*, 2008, pp. 165–174.

[38] H. Kim, S. Ko, D. S. Kim, and H. K. Kim, "Firewall ruleset visualization analysis tool based on segmentation," in *Proc. IEEE Symp. Visualizat. Cyber Secur. (VizSec)*, Oct. 2017, pp. 1–8.

[39] A. X. Liu and M. G. Gouda, "Complete redundancy detection in firewalls," in *Data and Applications Security XIX*, S. Jajodia and D. Wijesekera, Eds. Berlin, Germany: Springer, 2005, pp. 193–206.

[40] C. Basile, F. Valenza, A. Lioy, D. R. Lopez, and A. P. Perales, "Adding support for automatic enforcement of security policies in NFV networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 2, pp. 707–720, Apr. 2019.

[41] F. Valenza and M. Cheminod, "An optimized firewall anomaly resolution," *J. Internet Service Inf. Secur.*, vol. 10, no. 1, pp. 22–37, 2020.

[42] H. Levkowitz, *Color Theory and Modeling for Computer Graphics, Visualization, and Multimedia Applications* (The Springer International Series in Engineering and Computer Science). New York, NY, USA: Springer, 1997. [Online]. Available: https://books:google:co:kr/books?id=QwUMbEISxgQC

[43] R. B. Haber and D. A. McNabb, "Visualization idioms: A conceptual model for scientific visualization systems," *Vis. Sci. Comput.*, vol. 74, p. 93, Apr. 1990.

[44] R. S. Gallagher and S. Press, *Computer Visualization: Graphics Techniques for Engineering and Scientific Analysis*. Boca Raton, FL, USA: CRC Press, 1994.

[45] R. Likert, "A technique for the measurement of attitudes," *Arch. Psychol.*, vol. 140, no. 22, pp. 5–55, 1932.

[46] C. Diekmann, J. Michaelis, M. Haslbeck, and G. Carle, "Verified iptables firewall analysis," in *Proc. IFIP Netw. Conf. (IFIP Networking) Workshops*, May 2016, pp. 252–260.

[47] A. Saâdaoui, N. Ben Youssef Ben Souayeh, and A. Bouhoula, "FARE: FDD-based firewall anomalies resolution tool," *J. Comput. Sci.*, vol. 23, pp. 181–191, Nov. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877750317309894

[48] B. E. Logan and G. G. Xie, "Automating distributed firewalls: A case for software defined tactical networks," in *Proc. MILCOM - IEEE Mil. Commun. Conf. (MILCOM)*, Nov. 2019, pp. 1–6.

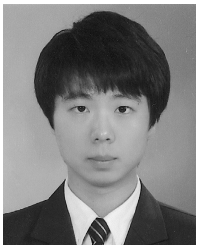
[49] S. Bagheri and A. Shameli-Sendi, "Dynamic firewall decomposition and composition in the cloud," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3526–3539, 2020.

TAEYONG KIM received the B.S. degree in information and telecommunication engineering from Mokwon University, Daejeon, South Korea, in 2012, and the M.S. degree in convergence science from the National Kongju University, Gongju, Chungnam, South Korea, in 2014.

From 2010 to 2011, he was a Researcher with Korea Institute of Science and Technology Information (KISTI), Daejeon, South Korea. From 2012 to 2019, he worked with a company related to Korea's network and information security. He is currently a Researcher with KISTI. His research interests include information security, network security, and data visualization.

TAEWOONG KWON received the B.S. degree in computer science and engineering from Soongsil University, Seoul, South Korea, in 2012, and the M.S. degree in information security from Korea University, Seoul, in 2014.

Since 2014, he has been a Researcher with Korea Institute of Science and Technology Information (KISTI), Daejeon, South Korea. His research interests include information security, network security, and data visualization.



JUN LEE received the B.S. and M.S. degrees in information and telecommunication engineering from Korea Aerospace University, South Korea, in 2010 and 2012, respectively, and the Ph.D. degree from the Graduate School, Korea Aerospace University, in 2017.

He worked as a Research Assistant with the Artificial Intelligence Cloud Research Team, National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan, from April 2017 to November 2017. From December 2017 to December 2019, he worked as a Postdoctoral Researcher with the Artificial Intelligent Research Center (AIRC), AIST, Tokyo. He is currently working as a Senior Researcher with the Science and Technology Cyber Security Center (S&T-CSC), Korea Institute of Science and Technology Information (KISTI). His research interests include knowledge discovering, natural language processing, data mining, and cyber security.

JUNGSUK SONG received the B.S. and M.S. degrees in information and telecommunication engineering from Korea Aerospace University, South Korea, in 2003 and 2005, respectively, and the Ph.D. degree from the Graduate School of Informatics, Kyoto University, Japan, in 2009.

From April 2009 to September 2010, he worked with the National Institute of Information and Communications Technology (NICT), Tokyo, Japan, as an Expert Researcher. From October 2010 to September 2011, he was a Researcher with the NICT. He is currently a Chief Researcher with Korea Institute of Science and Technology Information (KISTI), Daejeon, South Korea. He also serves as a Concurrent Professor with the University of Science and Technology (UST). His research interests include network security, artificial intelligence, data visualization, and security-related technologies.