IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 November 2023, accepted 18 December 2023, date of publication 25 December 2023,
date of current version 4 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3346922

RESEARCH ARTICLE

HSViz-II: Octet Layered Hierarchy Simplified Visualizations for Distributed Firewall Policy Analysis

HYUNJUNG LEE[1], SURYEON LEE[2], KYOUNGGON KIM[3], (Member, IEEE), AND HUY KANG KIM[4], (Member, IEEE)

[1]Korea Securities Computer Corporation (KOSCOM), Seoul 07329, South Korea
[2]Department of CyberSecurity, Seoul Women's University, Nowon, Seoul 01797, South Korea
[3]Center of Excellence in Cybercrime and Digital Forensics (CoECDF), Naif Arab University for Security Sciences, Riyadh 14812, Saudi Arabia
[4]School of Cybersecurity, Korea University, Seongbuk, Seoul 02841, South Korea

Corresponding author: Huy Kang Kim (cenda@korea.ac.kr)

This work was supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea government (MSIT) through the project ''Development of Intelligence Cyber Attack and Defense Analysis Framework for Increasing Security Level of C-ITS'' under Grant 2021-0-00624.

ABSTRACT Enterprises typically install firewalls at communication points to their internal networks with the primary objective of protecting their core assets from external cyber attackers. This ensures unauthorized access is controlled and prevented. However, overly permissive policies and services with vulnerabilities can be exploited by attackers, providing them with pathways into internal systems. Therefore, firewall policies must be meticulously applied and managed. Given the significant ramifications of firewall policies, they must be continuously managed with high importance. As the number of policies increases and the amount of information to be processed grows, the process becomes complex, and there are limitations to managing policies from a human cognitive perspective. An increase in unmanaged misuse policies can inadvertently introduce security risks through unintended allowance policies. In the case of large-scale network networks operating multiple firewalls, a different form of misuse policy check and management is required compared to managing a single firewall policy. The proposed tool, HSViz-II, not only visualizes misuse of a single firewall policy but also visualizes four misuse cases in a distributed firewall environment, providing a detailed breakdown based on Octets. It displays the distribution of anomalous policies by dividing the Source IP into Octet Layers. For the four anomalous policy cases, it offers five views based on dividing the Source IP into Octet Layers and three overall views for upstream firewall, downstream firewall, and both, totaling 60 views. The processing speed for each function was measured using four sets of actual upstream and downstream firewall policies, comprising eight different firewall policies in total. Firewall operators can use this tool to grasp the distribution status of misuse policies in single and multiple firewalls and check the status of misuse policies by Octet. By offering a method for firewall operators to accurately find meaningful information, this paper proposes a firewall misuse policy visualization system in a distributed firewall environment to help reduce the risk of asset exposure to cyber threats for enterprises. HSViz-II tool can be found on the web site: https://youtu.be/jvR8ZY2uapQ

INDEX TERMS Firewall policy visualization, policy analysis, data visualization, rule anomaly detection, distributed firewall rule anomaly detection.

I. INTRODUCTION

The need for in-depth discussions on solutions to policy conflict issues in distributed firewall environments, an extension from traditional single-firewall policy research, remains paramount. Not only in large-scale network infrastructures but also in small-scale network enterprises that deploy more than one firewall in a distributed manner, it is essential to prevent policy conflicts in distributed firewalls and optimize policies.

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

The reasons for systematizing and visualizing policy optimization tasks in a distributed firewall environment are threefold. Firstly, analyzing a complex firewall environment of a large-scale corporate network is limited by merely analyzing single firewall policies. As corporate networks expand, the complexity of increased traffic grows, and the number of firewalls traversed from source to destination for a service increases, making it challenging to analyze the interrelationships of distributed firewall policies. Detecting errors and optimizing policies by analyzing firewall policy correlations is a challenge faced by network and security administrators in practice. Understanding and visualizing the correlation and conflicts of distributed firewall policies aids in policy optimization and effective policy error detection. Secondly, it is essential to check excessively open or incorrectly opened policies in a distributed firewall environment to defend against cyber threats. A minor oversight by an administrator can lead to significant risks, and systematizing policy analysis for easy visualization can mitigate this. Thirdly, it aims to reduce latency in firewall traffic processing and ensure stable availability. An increase in the number of firewall policies can adversely affect the firewall's traffic processing performance. For this reason, it's necessary to clean up redundant and misuse policies and optimize them. Policy misuse detection in a single firewall environment involves understanding the interrelationships of each policy applied to one firewall, while in a distributed firewall environment, it involves checking issues arising from misuse as a policy applied to one firewall passes through another. Therefore, policies applied to a single firewall and those applied to multiple firewalls differ in misuse form and type, depending on whether the misuse occurs in one firewall or as it passes through several firewalls.

This paper upgrades the Hierarchy Simplified View covered in the tool HSViz from previous studies to detect misuse policies in a distributed firewall environment. While 3D charts have the advantage of conveying compressed information, they can miss details due to their complexity when one wants to find firewall rule errors in detail. There can also be speed and performance constraints when plotting in 3D for a wide IP address space. For reducing firewall policy complexity and showing misuse policy distribution by IP octet, a heatmap is optimal for visualization chart design. This visualization tool aids in quickly and easily discerning the four types of firewall policy misuses in a distributed firewall environment. The contributions of our paper are summarized as follows.

- The distribution of misuse policies in both single and distributed firewall environments can be identified by IP octet.
- Network and security administrators, even those with limited expertise in networking and security, can intuitively detect anomalous policies with ease.
- Reducing complexity and lowering computational costs, it provides intuitive visualization results.

The rest of the paper is organized as follows. Chapter 2 delves into the necessity of visualizing firewall policies and reviews pertinent research. Chapter 3 provides a detailed explanation of the four cases of firewall misuse policies. In Chapter 4, the proposed tool, HSViz-II, and its design and specific features are introduced. Chapter 5 outlines potential applications of the tool. Chapter 6 conducts a comprehensive evaluation of the tool, and Chapter 7 concludes the study, shedding light on future research directions.

II. BACKGROUND

A. NEED FOR FIREWALL POLICY VISUALIZATION

Firewalls play a pivotal role in safeguarding internal systems by either blocking or permitting data entering from external sources and data exiting to external destinations. They are instrumental in preventing data exfiltration, blocking malicious codes, and detecting anomalies. The management of firewall policies is of paramount importance from the perspectives of bolstering security and preventing data leaks. As the scale and complexity of corporate networks grow, the number and intricacy of firewall policies increase correspondingly. Analyzing such vast and intricate data manually has its limitations, making visualization a valuable tool. Visualization refers to the process of graphically representing data and information, facilitating a more intuitive understanding. Through visualization, we can easily comprehend intricate data, discern patterns and flows in data, and swiftly grasp the interconnections between data sets. This is crucial for understanding complex data and discerning relationships among them. Visualizing firewall policies enables firewall operators or security administrators to instantly verify policy usage and grasp the flow of policies at a glance. This heightened awareness of policies empowers security administrators to respond more promptly to security threats.

B. RELATED WORK

Research on the visualization of firewall policies can be primarily segmented into three distinct categories:

- Analytical Visualization of Configured Policies: This focuses on the analysis and subsequent visualization of policies that are already established within the firewall.
- Visualization of Policy Misuse: This domain emphasizes the effective visualization of misappropriated firewall policies, aiming to visually address these misuses.
- Policy Visualization in distributed Firewall Environments: This category explores the challenges and methodologies associated with visualizing policies in environments that incorporate distributed firewalls.

1) FIREWALL POLICY VISUALIZATION

Hyungseok Kim et al. [1] implemented firewall policy visualization in a 6-dimensional space using 3D graphics through a tool named FRuVATS (Firewall Ruleset Visualization Analysis Tool based on Segmentation). This tool is capable of representing the entire range of source and destination IPs. It analyzes policies where conflicts arise, identifies active and inactive areas, and displays the identified results using a visualization model. Mansmann et al. [2] proposed a tool called VISUAL FIREWALL EDITOR. Utilizing the Sunburst Chart, it represents the status of firewall policy ACL (access-list) and objects and groups. A distinctive feature of this tool is its ability to identify host objects based on network and group objects. Morrissey et al. [3] introduced a tool named Created Voids. This tool analyzes firewall policies and, using a parallel coordinate graph, differentiates between areas that are permitted and those that are not. The tool visualizes the status of firewall policies, whether they are allowing or denying, in a 3D format on the parallel coordinate graph, enhancing visual perception.

2) ABUSE POLICY RESOLUTION AND VISUALIZATION

Ehab Al-Shaer et al. [4], [5], [6] conducted research on the detection of anomalous rules in firewall policies. They categorized misuse policies into four types: Shadowing, Generalization, Redundancy, and Correlation. They proposed an algorithm to detect these misuses and utilized the Policy Advisor for their identification. Kim et al. [7] employed the N-ary tree module for the swift inspection and detection of a vast number of rules pertaining to firewall misuse policies. They presented the detection results through 3D visualization. Hu et al. [8] introduced a tool named Fame. This tool identifies and addresses policy misuse by visualizing rule-based segmentation techniques using a grid-based representation. FIREMAN [9] conducts symbolic model checking for all IP packets and paths. It models firewall rules using Binary Decision Diagrams (BDD), a technique that has been successfully applied in hardware verification and model validation. Saâdaoui et al. [10] attempted to resolve firewall policy misuse issues using Firewall Decision Diagrams (FDD). They automated the process of optimizing and organizing rules by eliminating unnecessary ones from the firewall, detecting, and rectifying misconfigurations. Chao and Yang [11] proposed a BST-based bit-vectorization method for calculating firewall rule space in the IPv6 system, which has a 128-bit address structure. They vectorized the spatial relationship between filtering rules by encoding them and analyzed policy misuse through FSM-based comparisons. In the traffic filtering diagram, rules without intersections are eliminated, and the remaining ones undergo detailed analysis. The spatial relationship in the traffic filtering diagram between rules is computed, and the bit-vector is input into the FSM to obtain the final diagnostic result. These results are visualized as a 2D traffic filtering diagram, representing blocks of vectors, SIP, DIP durations, and all rules affecting the block. The visualization offers 360-degree rotation, zoom in/out, rule selection, and 3D representation. Kim et al. [12] proposed a 3D-based hierarchical visualization tool named F/Wvis. Notably, F/Wvis offers a drill-down user interface through a hierarchical visualization approach, supporting ACL management for large-scale networks and detailed anomaly analysis of policies.

3) VISUALIZE DISTRIBUTED FIREWALL POLICIES

Hazem Hamed and Ehab Al-Shaer [13] proposed an approach to analyze firewall policies utilizing Relational Algebra and the 2D-Box model. They articulated the relationships between firewall policies using relational algebra operations and introduced an effective cognition strategy for four instances of firewall policy misuse via the Raining 2D-Box model. The authors proposed methods for policy optimization by eliminating misuse policies and combining policies, and they introduced a program to extract misuse policies. The 2D-Box similarity model presented in their study was later employed in a paper by Yu-Zhu Cheng1 and Qiu-ying Shi to compare distributed firewall policies. Using the Raining 2D-Box model, they identified misuses such as overlaps in policies between upstream and downstream firewalls and conducted simulations. By applying the Raining 2D-Box model in a distributed firewall setting, they derived an algorithm to detect two types of misuses: Shadowing anomaly and Superiousness anomaly. The paper also showcased the results of the time taken by the algorithm to detect these anomalies. Chao [14] employs a three-tiered visualization hierarchy, segmented into physical, logical, and misbehavior views. This stratified approach offers users visualization results concerning the relationships between firewalls, aiding in firewall policy error debugging and the removal of misuse policies. Pisharody et al. [15] introduce Brew, a security policy analysis framework implemented in the OpenDaylight SDN controller. In a distributed SDN-based cloud environment, they ensure the implementation of consistent, collision-free security policies by guaranteeing that two rules do not conflict at any layer, thereby preventing information leakage. They present a technique for assigning global priorities to flow rules in a distributed setting. Recognizing and categorizing collisions arising from cross-layer conflicts, they extend firewall rule collision categorization to SDN flow rule collisions in traditional settings. Furthermore, they propose strategies to resolve these collisions without external support.

C. FIREWALL POLICY OVERVIEW

Figure 1 schematically represents the relationships among firewall policies. These relationships can be categorized as Inclusively Matched (IM), Exactly Matched (EM), Partially Matched (PM), Completely Disjoint (CD), and Correlated. The Inclusively Matched (IM) relationship is defined when one firewall policy is encompassed within a portion of another policy. Given the existence of policies Rx and Ry, if the entirety of Rx is contained within a segment of Ry and the sequence of policy Rx is lower than that of Ry, then Rx is deemed a superfluous and redundant misuse policy. The Exactly Matched (EM) relationship arises when two firewall

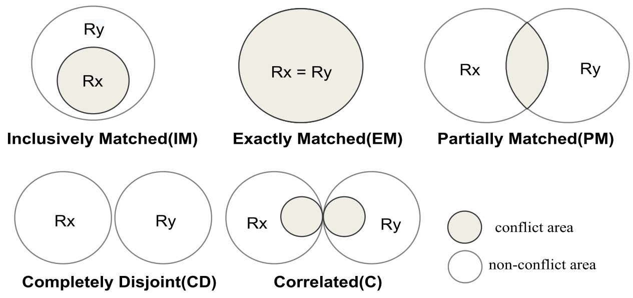


FIGURE 1. Firewall policy relation.

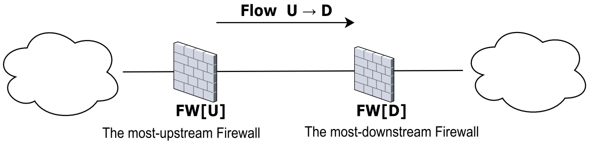


FIGURE 2. Upstream and downstream firewall concept [6].

TABLE 1. FW[U], examples of firewall policies in a packet filter firewall.

Order	Protocol	Source IP	Destination IP	Port	Action
1	TCP	140.192.37.2	161.120.33.40-46	25	deny
2	TCP	140.192.37.0-10	161.120.33.42	20	deny
3	TCP	140.192.37.0-10	161.120.33.41	25	allow
4	TCP	140.192.37.1	161.120.33.41	25	deny
5	TCP	140.192.37.3	161.120.33.43	21	allow
6	TCP	140.192.37.0-10	161.120.33.43	21	deny
7	TCP	140.192.37.1	161.120.33.42	20	deny
8	TCP	140.192.37.0-15	162.120.33.43	50	allow
9	TCP	140.192.37.0-10	162.120.33.43	50	deny
10	TCP	161.120.33.5-7	140.192.37.0-10	21,80	deny
11	TCP	161.120.33.0-10	140.192.37.4-7	25	allow
12	TCP	161.120.33.5-7	140.192.37.4-7	21,80	allow
13	TCP	161.120.33.5-7	140.192.37.1-2	25	deny

policies are identically aligned. Specifically, if policies Rx and Ry are present and Rx is equivalent to Ry, the policy with the inferior sequence is considered a redundant misuse policy. The Partially Matched (PM) relationship describes a scenario where segments of two firewall policies overlap. For instance, if a segment of firewall policy Rx overlaps with a portion of firewall policy Ry, there's a need to streamline the overlapping segment of the firewall policies. The Completely Disjoint (CD) relationship is characterized when two firewall policies do not overlap and are distinct from each other. The Correlated (C) relationship is defined when parts of two firewall policies overlap, but the content of the overlapping region differs between the policies. For example, it signifies a scenario where an overlapping region exists between two firewall policies, Rx and Ry. In a distributed firewall environment, we conducted an analysis and visualization of each anomaly policy based on the firewall policies in Figure 2 and Tables 1 and 2.

TABLE 2. FW[D], examples of firewall policies in a packet filter firewall.

Order	Type	SIP	DIP	Port	Action
1	TCP	161.120.33.5-7	140.192.37.4-7	80	allow
2	TCP	161.120.33.5-7	140.129.37.5	21	deny
3	TCP	161.120.33.0-10	140.192.37.4-7	21	deny
4	TCP	161.120.33.6	140.192.37.4-7	23	allow
5	TCP	161.120.33.7	140.192.37.1-10	25	deny
6	TCP	161.120.33.5-7	140.192.37.2-6	25	allow

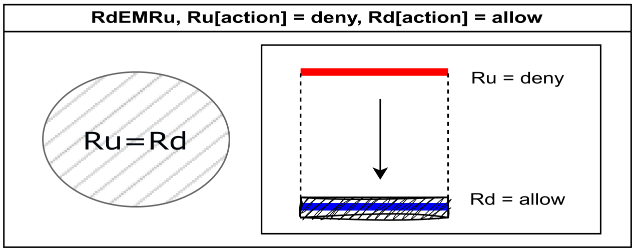


FIGURE 3. Shadowing anomaly example 1.

TABLE 3. Firewall policy: shadowing anomaly example 1.

Name	Type	SIP	DIP	DPort	Action
Ru	21	161.120.*.*	140.192.22.5	21	deny
Rd	21	161.120.*.*	140.192.22.5	21	allow

Anomaly policies in distributed firewalls can be broadly categorized into shadowing, spuriousness, redundancy, and correlation. The correlation between the four anomaly policy cases has been schematically represented using Overlapping and Action. Ru denotes the upstream firewall policy, while Rd represents the downstream firewall policy. Detailed examples of the four anomaly policy scenarios have been illustrated for easy comprehension. Deny policies for Ru and Rd are marked in red, while allow policies are indicated in blue. Policies that directly cause misuse are highlighted with a cross-hatched box for easy identification.

1) SHADOWING ANOMALY

The Shadowing Anomaly refers to a rule that can produce unintended outcomes due to overlapping policies within a firewall. Such policies can lead to significant errors, especially when the policy actions are contradictory, resulting in a rule that should be permitted being denied. Such policies should either be removed or reordered. If the Upstream firewall policy and the Downstream firewall policy are inclusive of each other but have conflicting policy actions, or if they are entirely identical but have different policy actions, or if the firewall policies are inclusive of each other with the same policy actions, a Shadowing Anomaly occurs.

$$RdEMRu,\ Ru\,[action] = deny,\ Rd\,[action] = allow$$
$$RdIMRu,\ Ru\,[action] = deny,\ Rd\,[action] = allow$$
$$RuIMRd,\ Ru\,[action] = deny,\ Rd\,[action] = allow$$
$$RuIMRd,\ Ru\,[action] = allow,\ Rd\,[action] = allow$$

The policy misuse scenarios have been illustrated based on different situations. As shown in the Figure 3 and Table 3, when the upstream and downstream firewall policies are completely identical but have different policy actions, the downstream firewall policy is shadowed.

As shown in Figure 4 and Table 4, when the upstream firewall policy encompasses the downstream firewall policy

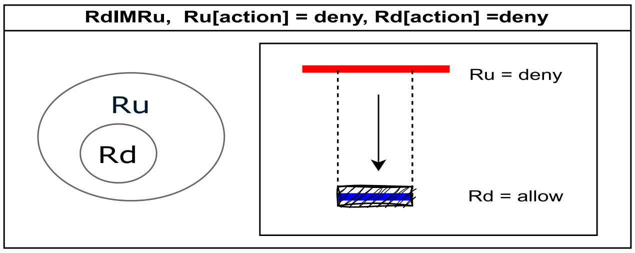


FIGURE 4. Shadowing anomaly example 2.

TABLE 4. Firewall policy: shadowing anomaly example 2.

Name	Type	SIP	DIP	Port	Action
Ru	TCP	*.*.*.*	*.*.*.*	any	deny
Rd	TCP	*.*.*.*	161.120.33.*	23	allow

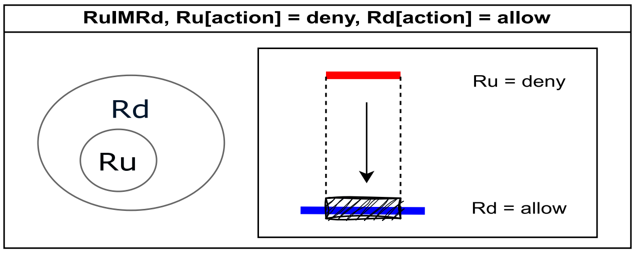


FIGURE 5. Shadowing anomaly example 3.

TABLE 5. Firewall policy: shadowing anomaly example 3.

Name	Type	SIP	DIP	Port	Action
Ru	TCP	161.120.24.*	140.192.22.5	25	deny
Rd	TCP	161.120.24.*	140.192.*.*	25	allow

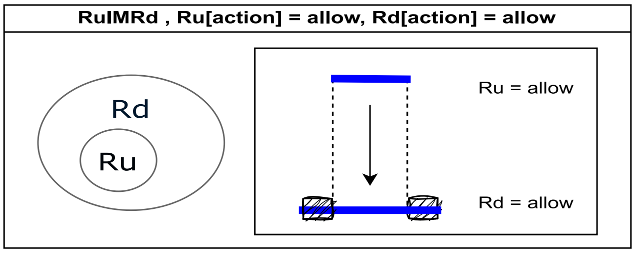


FIGURE 6. Shadowing anomaly example 4.

and their policy actions are distinct, parts of the downstream firewall policy are shadowed by the upstream firewall policy.

As illustrated in Figure 5 and Table 5, when the upstream firewall policy encompasses the downstream firewall policy and their policy actions differ, parts of the downstream firewall policy are shadowed by the upstream firewall policy.

As shown in Figure 6 and Table 6, when the upstream firewall policy encompasses the downstream firewall policy and their policy actions are identical, parts of the downstream firewall policy are shadowed.

TABLE 6. Firewall policy: shadowing anomaly example 4.

Name	Type	SIP	DIP	Port	Action
Ru	TCP	161.120.33.*	140.192.*.*	23	allow
Rd	TCP	161.120.*.*	140.192.*.*	23	allow

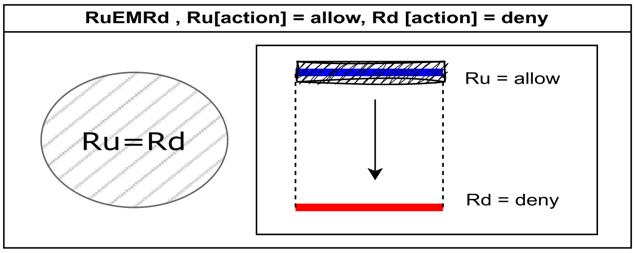


FIGURE 7. Spuriousness anomaly example 1.

TABLE 7. Firewall policy: spuriousness anomaly example 1.

Name	Type	SIP	DIP	Port	Action
Ru	TCP	140.192.*.*	161.120.*.*	80	allow
Rd	TCP	140.192.*.*	161.120.*.*	80	deny

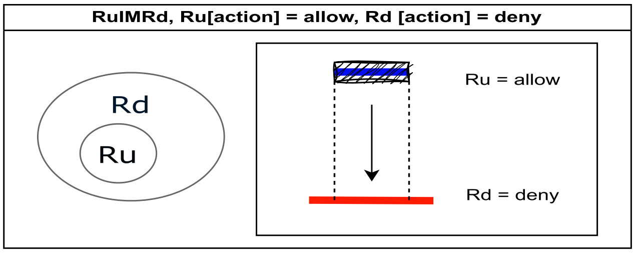


FIGURE 8. Spuriousness anomaly example 2.

2) SPURIOUSNESS ANOMALY

A spuriousness anomaly occurs when a policy blocked by the downstream firewall is allowed by the upstream firewall.

$$\begin{aligned} &RuEMRd, Ru\,[action] = allow, Rd\,[action] = deny\\ &RuIMRd, Ru\,[action] = allow, Rd\,[action] = deny\\ &RdIMRu, Ru\,[action] = allow, Rd\,[action] = deny\\ &RdIMRu, Ru\,[action] = allow, Rd\,[action] = allow\\ &RuIMRd, Ru\,[action] = deny, Rd\,[action] = deny \end{aligned}$$

As shown in the Figure 7 and Table 7, when the upstream firewall policy matches the downstream firewall policy, and the action in the upstream firewall is set to ‘‘allow’’ while the downstream firewall is set to ‘‘block’’, a policy error occurs in the downstream firewall policy.

As shown in the Figure 8 and Table 8, when the upstream firewall policy encompasses the downstream firewall policy, and the action in the upstream firewall is set to ‘‘allow’’ while the downstream firewall is set to ‘‘block’’, an error arises in the allowed upstream firewall policy.

As shown in the Figure 9 and Table 9, when the downstream firewall policy is encompassed by the upstream firewall policy, and the upstream firewall policy is set to ‘‘allow’’ while the downstream firewall policy is set to

TABLE 8. Firewall policy: spuriousness anomaly example 2.

Name	Type	SIP	DIP	Port	Action
Ru	TCP	140.192.*.*	161.120.*.*	80	allow
Rd	TCP	*.*.*.*	*.*.*.*	any	deny

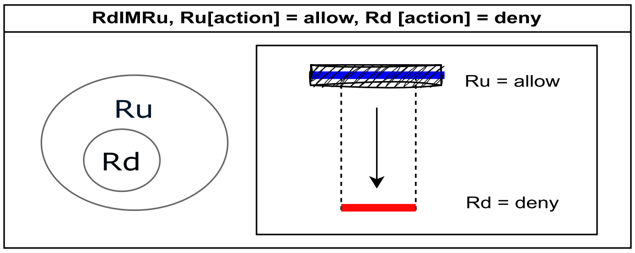


FIGURE 9. Spuriousness anomaly example 3.

TABLE 9. Firewall policy: spuriousness anomaly example 3.

Name	Type	SIP	DIP	Port	Action
Ru	TCP	161.120.33.*	140.192.*.*	23	allow
Rd	TCP	161.120.33.*	140.192.37.*	23	deny

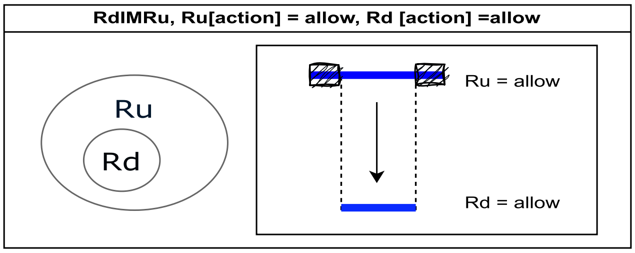


FIGURE 10. Spuriousness anomaly example 4.

TABLE 10. Firewall policy: spuriousness anomaly example 4.

Name	Type	SIP	DIP	Port	Action
Ru	TCP	161.120.*.*	140.192.*.*	21	allow
Rd	TCP	161.120.*.*	140.192.22.5	21	allow

TABLE 11. Firewall policy: spuriousness anomaly example 5.

Name	Type	SIP	DIP	Port	Action
Ru	TCP	161.120.33.*	140.192.37.1	23	deny
Rd	TCP	140.192.*.*	161.120.33.*	23	allow

"block", an error occurs in a portion of the upstream firewall policy.

As shown in the Figure 10 and Table 10, when the downstream firewall policy is encompassed by the upstream firewall policy, and both the upstream and downstream firewall policies are set to "allow", an error arises in a portion of the upstream firewall policy.

As shown in the Figure 11 and Table 11, when the upstream firewall policy encompasses the downstream firewall policy, and both the upstream and downstream firewall policies are set to "deny", an error arises in a portion of the downstream firewall policy.

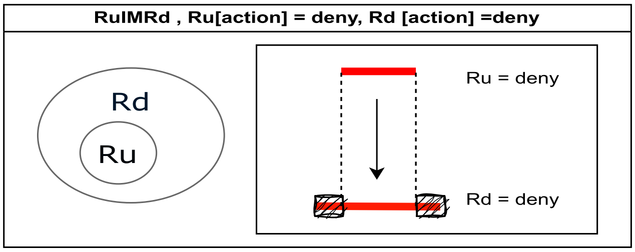


FIGURE 11. Spuriousness anomaly example 5.

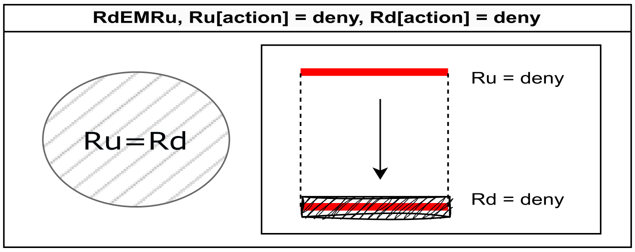


FIGURE 12. Redundancy anomaly example 1.

TABLE 12. Firewall policy: redundancy anomaly example 1.

Name	Type	SIP	DIP	Port	Action
Ru	TCP	161.120.24.*	140.192.37.3	25	deny
Rd	TCP	161.120.24.*	140.192.37.3	25	deny

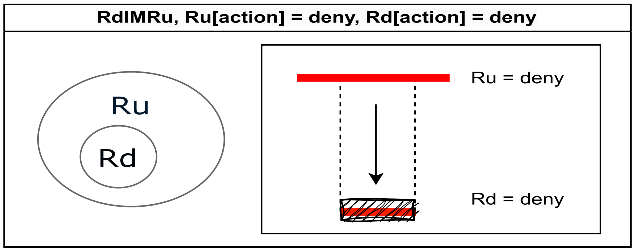


FIGURE 13. Redundancy anomaly example 2.

3) REDUNDANCY ANOMALY

Redundancy Anomaly refers to the presence of duplicated firewall policies. This not only complicates policy management but can also lead to unintended security threats.

$$RdEMRu, Ru\ [action] = deny, Rd\ [action] = deny$$
$$RdIMRu, Ru\ [action] = deny, Rd\ [action] = deny$$

As shown in the Figure 12 and Table 12, when the downstream firewall policy matches the upstream firewall policy and both have a "deny" action, a Redundancy Anomaly occurs due to the duplication of policies.

As shown in the Figure 13 and Table 13, when the downstream firewall policy is encompassed by the upstream firewall policy and both have a "deny" action, a Redundancy Anomaly arises.

4) CORRELATION ANOMALY

The Correlation anomaly occurs when firewall policies intersect, either partially matching or encompassing one another. This makes it challenging to easily discern the

TABLE 13. Firewall policy: redundancy anomaly example 2.

Name	Type	SIP	DIP	Port	Action
Ru	TCP	*.*.*.*	*.*.*.*	any	deny
Rd	TCP	161.120.*.*	140.192.*.*	22	deny

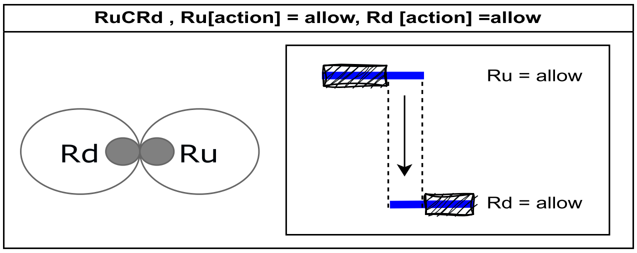


FIGURE 14. Correlation anomaly example 1.

TABLE 14. Firewall policy: correlation anomaly example 1.

Name	Type	SIP	DIP	Port	Action
Ru	TCP	140.192.*.*	161.120.33.*	80	allow
Rd	TCP	140.192.37.*	161.120.*.*	80	alow

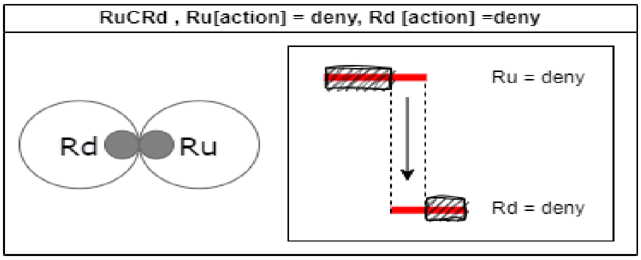


FIGURE 15. Correlation anomaly example 2.

TABLE 15. Firewall policy: correlation anomaly example 2.

Name	Type	SIP	DIP	Port	Action
Ru	TCP	140.192.*.*	161.120.33.*	80	deny
Rd	TCP	140.192.37.*	161.120.*.*	80	deny

firewall policies and can result in policies shadowing each other partially.

$$RdCRu, Ru\ [action] = allow, Rd\ [action] = allow$$

$$RdCRu, Ru\ [action] = deny, Rd\ [action] = deny$$

$$RdCRu, Ru\ [action] = allow, Rd\ [action] = deny$$

$$RdCRu, Ru\ [action] = deny, Rd\ [action] = allow$$

As shown in the Figure 14 and Table 14, when the downstream firewall policy is mutually inclusive with the upstream firewall policy, and their actions are identical, a Correlation Anomaly error occurs. Due to the correlation of the policies, parts of both the upstream and downstream firewall policies are not permitted.

As shown in the Figure 15 and Table 15, when the downstream firewall policy is mutually inclusive with the upstream firewall policy and their actions are identical, a Correlation Anomaly error arises.

As shown in the Figure 16 and Table 16, when the downstream firewall policy is mutually inclusive with the upstream firewall policy and their actions differ, a Correlation Anomaly error arises.

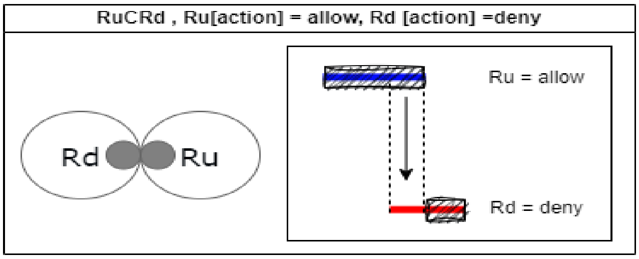


FIGURE 16. Correlation anomaly example 3.

TABLE 16. Firewall policy: correlation anomaly example 3.

Name	Type	SIP	DIP	Port	Action
Ru	TCP	140.192.*.*	161.120.33.*	80	deny
Rd	TCP	140.192.37.*	161.120.*.*	80	alow

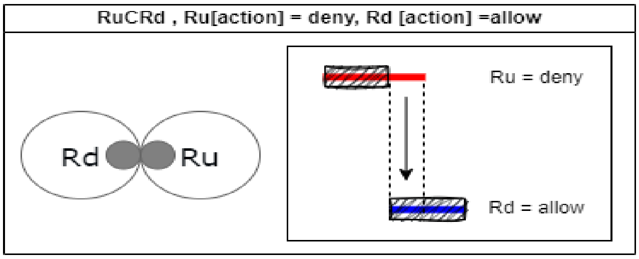


FIGURE 17. Correlation anomaly example 4.

TABLE 17. Firewall policy:correlation anomaly example 4.

Name	Type	SIP	DIP	Port	Action
Ru	TCP	161.120.*.*	140.192.22.5	21	deny
Rd	TCP	161.120.*.*	140.192.22.5	21	allow

As shown in the Figure 17 and Table 17, when the downstream firewall policy intersects with the upstream firewall policy and their actions are inconsistent, a Correlation Anomaly error occurs.

III. VISUALIZATION TOOL OVERVIEW

A. BASIC DESIGN

In the original HSViz design concept, the HierarchyView was introduced to visualize firewall policies by segmenting them based on IP Octets. This approach allowed users to intuitively perceive the distribution of policies based on Octets, highlighting the presence or absence of policies for selected source and destination IPs in a grid format. By segmenting the source IP based on Octets and providing users with selection options, they can choose their desired view. Users can identify the presence of policies for the source and destination IPs through distinct colors or shapes in the grid. By refining the representation area based on the selected A, B, C Octets of the existing policy's source IP, users can verify the details and distribution of the policies associated with the selected IP. The X-axis represents the source address, and the Y-axis represents the destination address, allowing

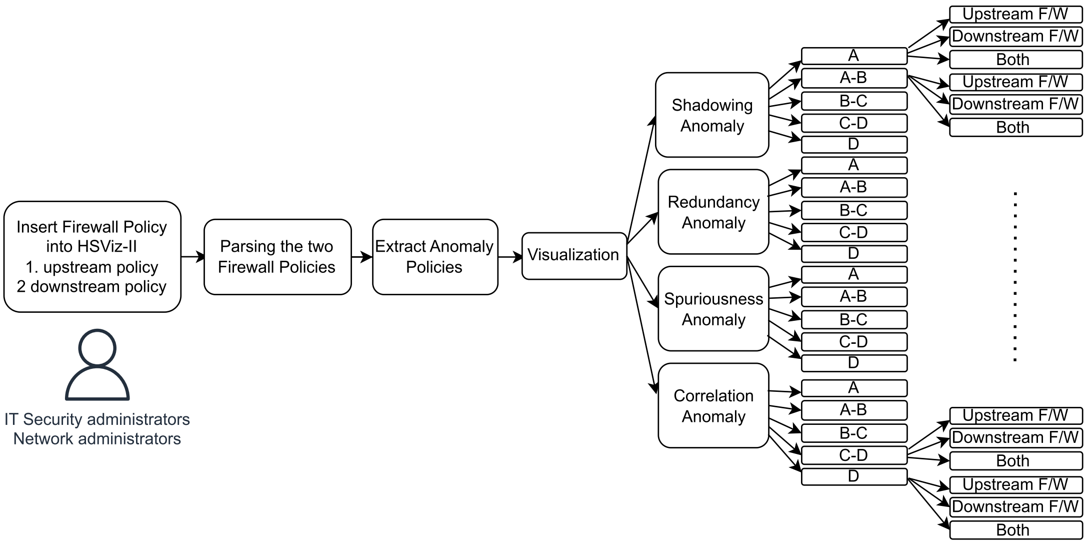


FIGURE 18. HSViz-II Process Flow.

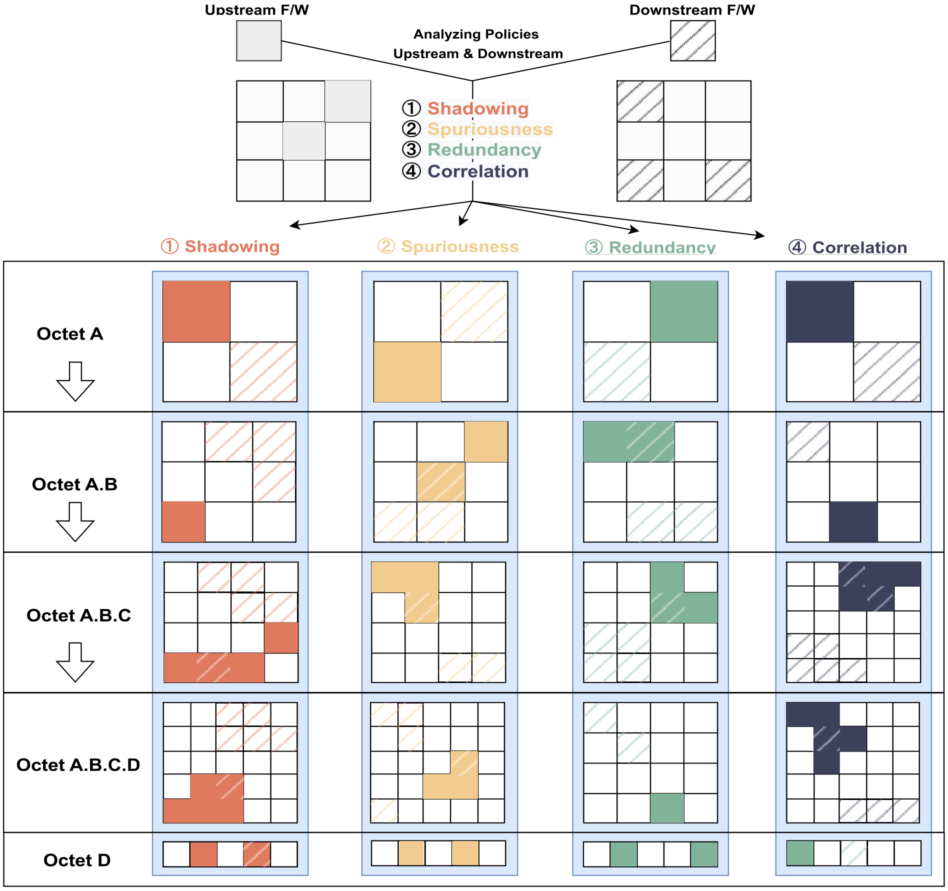
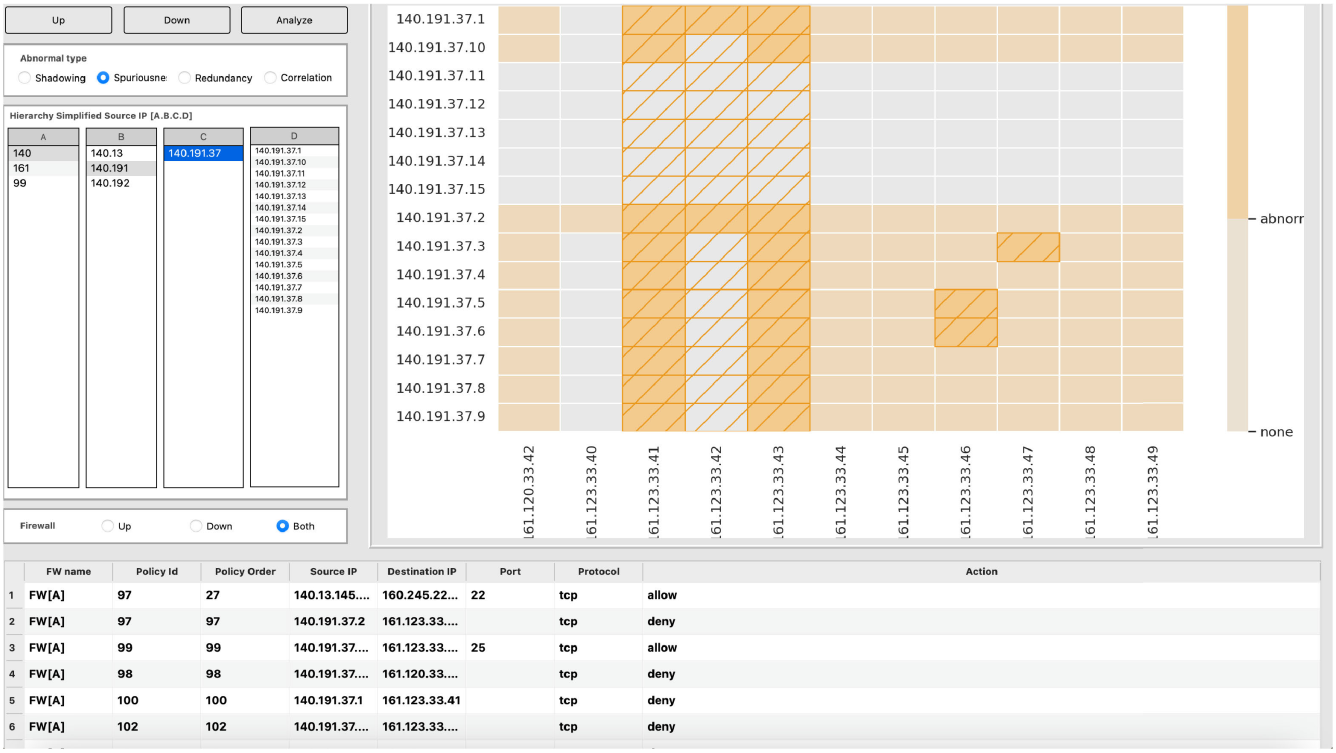


FIGURE 19. HSViz-II Design Concept.

users to check the presence of policies based on the selected source address in an IP Octet-based hierarchy.

As shown in the Figure 18, HSViz-II takes as input the upstream firewall and downstream firewall policies. It then parses these two policies to extract any anomalous policies. Following this, through a visualization process, it categorizes the anomalies into four misuse cases. The system then displays these cases across five different Octet



Abnormal type
○ Shadowing ● Spuriousne: ○ Redundancy ○ Correlation

Hierarchy Simplified Source IP [A.B.C.D]

A	B	C	D
140 161 99	140.13 140.191 140.192	140.191.37	140.191.37.1 140.191.37.10 140.191.37.11 140.191.37.12 140.191.37.13 140.191.37.14 140.191.37.15 140.191.37.2 140.191.37.3 140.191.37.4 140.191.37.5 140.191.37.6 140.191.37.7 140.191.37.8 140.191.37.9

Firewall ○ Up ○ Down ● Both

	FW name	Policy Id	Policy Order	Source IP	Destination IP	Port	Protocol	Action
1	FW[A]	97	27	140.13.145....	160.245.22...	22	tcp	allow
2	FW[A]	97	97	140.191.37.2	161.123.33....		tcp	deny
3	FW[A]	99	99	140.191.37....	161.123.33....	25	tcp	allow
4	FW[A]	98	98	140.191.37....	161.120.33....		tcp	deny
5	FW[A]	100	100	140.191.37.1	161.123.33.41		tcp	deny
6	FW[A]	102	102	140.191.37....	161.123.33....		tcp	deny

FIGURE 20. HSViz-II overall view.

layers: A-Octet, A-B Octet, B-C Octet, C-D Octet, and D-Octet. For each of these Octet layers, HSViz-II provides three distinct views: the Upstream firewall view, the Downstream firewall view, and the Both view, which showcases policies from both firewalls simultaneously.

Taking the HierarchyView design concept from the original HSViz a step further, As shown in the Figure 19 and Figure 21, it has been applied to the analysis of distributed firewall policies, enhancing the visualization format. The system presents a hierarchical view of misuse policies derived from the upstream firewall policy FW[U] and the downstream distributed firewall policy FW[D]. After analyzing the firewall policy FW[U] and FW[D], the system extracts four predefined misuse cases. Users can select the desired Octet layer of the source IP to view the status of misuse policies in a refined Octet-based hierarchy. Misuse policies from FW[U] are represented in various colors, while those from FW[D] are represented with diagonal stripes. Furthermore, misuse policies present in both FW[U] and FW[D] are represented with colored boxes with diagonal stripes.

The essence of the proposed system is its ability to provide a granular or holistic view of policy misuse distribution in a distributed firewall environment based on IP Octets. By offering a new perspective on distributed firewall policies that were previously unseen, the system holds significant value. The segmentation of policies for visualization reduces complexity, and the grid format allows users to intuitively understand the details. By providing users with selection options based on Octet-segmented source IPs, the system simplifies complex IP configurations, enabling users to verify details. The visual information provided to users in the grid format includes source and destination information.

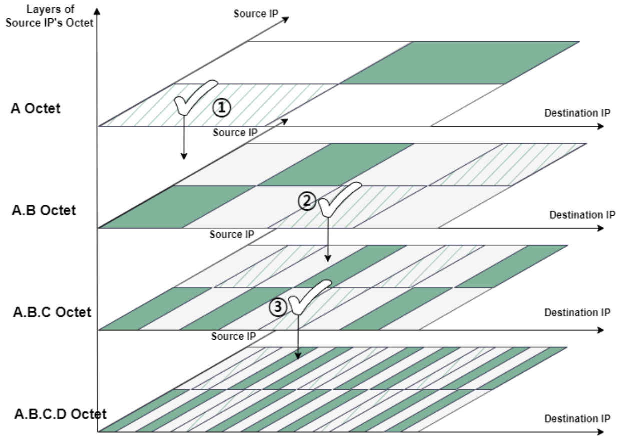


FIGURE 21. Octet layered anomaly view.

B. OVERALL VIEW

As depicted in the Figure 20, the visualization results are presented in three distinct scenarios based on the firewall policies: the upstream firewall policy Fw[U], the downstream firewall policy Fw[D], and the Both case, which displays both firewall policies simultaneously. These scenarios are further categorized based on the Octet structure into five

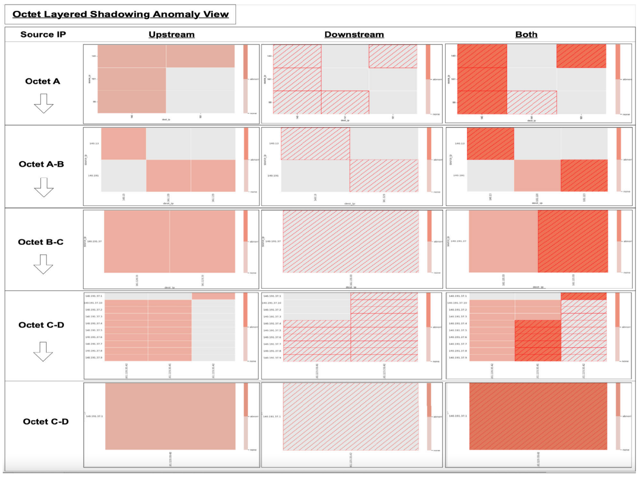


FIGURE 22. Layered octet based shadowing anomaly view.

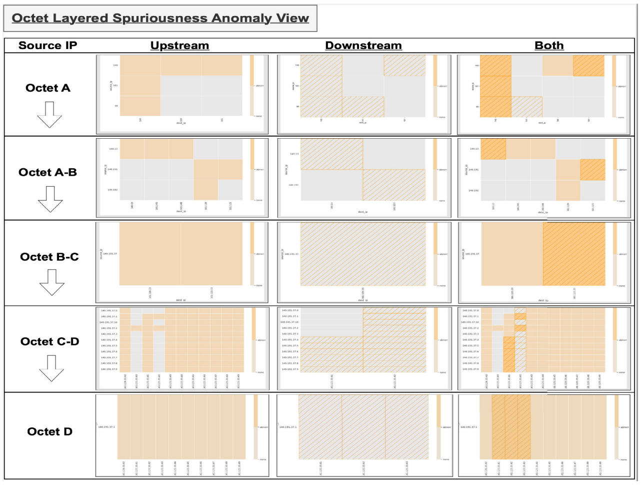


FIGURE 23. Layered octet based spuriousness anomaly view.

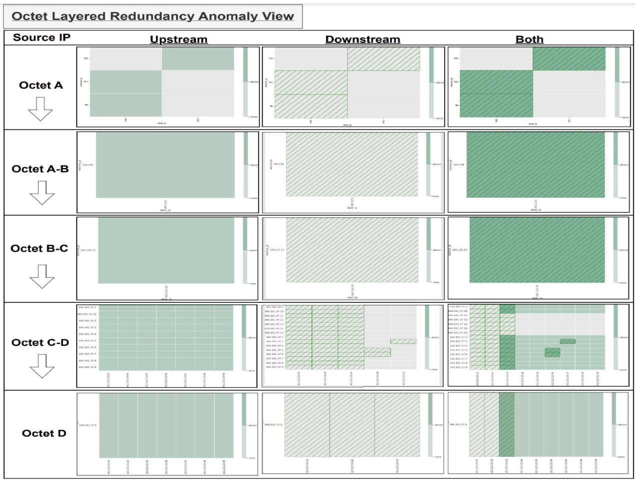


FIGURE 24. Layered octet based redundancy anomaly view.

segments: A, A.B, A.B.C, A.B.C.D, and D, resulting in a total of 15 different views.

Users have the option to choose which firewall view they wish to inspect, whether it's Upstream, Downstream, or Both. If they select the Abnormal type "Shadowing", the system will analyze the two firewall policies to identify the Shadowing Anomaly policies. Subsequently, users can select their desired Octet layer based on the Source IP to examine the distribution of the policies.

The Upstream view highlights the misuse policies found in the upstream firewall policy, while the Downstream view does the same for the downstream firewall policy. The Both view simultaneously displays misuse policies from both firewalls.

This approach offers a comprehensive visualization of the entire set of firewall policies across the Upstream, Downstream, and Both scenarios, broken down by each Octet. This granularity is especially useful when users want to delve deep into the distribution of policies across each Octet layer. This innovative approach, which was previously unavailable, provides operators with valuable insights through enhanced visualization, making it a significant advancement in the field.

C. ANOMALY VIEW

1) SHADOWING ANOMALY

The visualization results for the Shadowing Anomaly are derived based on the four scenarios introduced in II-C-1. These anomalies are categorized as Shadowing Anomalies and are visualized based on the Upstream, Downstream, and Both policies. Furthermore, the visualization is segmented based on the four Octet layers of the Source IP. As shown in the Figure 22, a total of 15 different views are derived.

2) SPURIOUSNESS ANOMALY

The visualization results for Spuriousness Anomaly derive from the five scenarios presented in II-C-2 and are categorized as Spuriousness Anomaly. These results are visualized based on the Upstream, Downstream, and Both policies, segmented by the four octets of the Source IP as layers. As depicted in the Figure 23, a total of 15 visual representations are generated.

3) REDUNDANCY ANOMALY

The visualization results for Redundancy Anomaly are derived from the two scenarios presented in II-C-3 and are categorized as Redundancy Anomaly. These results are visualized based on the Upstream, Downstream, and Both policies, segmented by the four octets of the SourceIP as layers. As depicted in the Figure 24, a total of 15 visual representations are generated.

4) CORRELATION ANOMALY

The visualization results for Correlation Anomaly are derived from the four scenarios presented in II-C-4 and are categorized as Correlation Anomaly. These results are visualized based on the Upstream, Downstream, and Both policies, segmented by the four octets of the SourceIP as

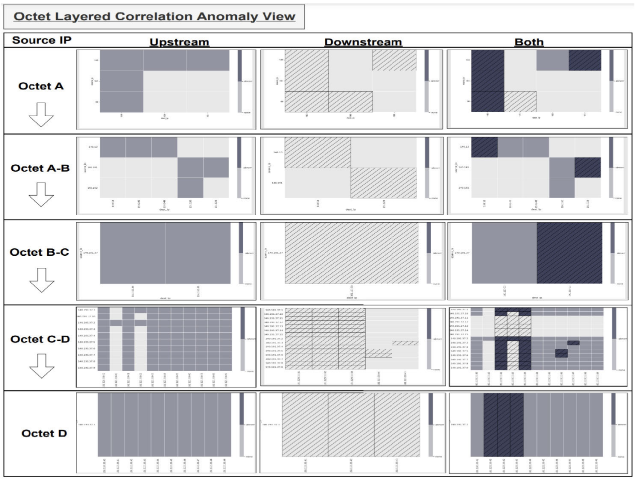


FIGURE 25. Layered octet based correlation anomaly view.

layers. As illustrated in the Figure 25, a total of 15 visual representations are generated.

IV. USAGE SCENARIO

The HSViz-II system presented in this paper was designed considering the utilization needs of firewall operators, security auditors, and policy decision-makers. By using this tool, one can easily grasp the distribution status of firewall misuse policies. It aids in communication with non-practitioners and assists in making decisions regarding policy improvements, ultimately contributing to the secure operation of firewalls.

A. USED TO IMPROVE ABNORMAL POLICIES FOR EACH SERVICE

In organizations with large-scale network infrastructures, different octets of IP addresses are set and assigned to services, and the teams and organizational structures responsible for these services differ. For instance, Task A might be serviced under the 10.0.0.0/8 range, while Task B is under the 172.0.0.0/8 range. These IPs can be further subdivided by octet to allocate more granularly for specific services. In such environments, using the aforementioned tool allows one to verify the distribution of misuse policies based on octets. This facilitates easy identification of which tasks or services have a high occurrence of misuse policies. Consequently, firewall operators can intuitively identify the responsible parties or stakeholders for discussions on the status and potential improvement measures for misuse policies. Additionally, by being able to ascertain the distribution of misuse policies, prioritized action plans can be established for tasks where a significant number of misuse policies have been detected.

B. CHECKING MISUSE POLICY DISTRIBUTION IN A MULTI-FIREWALL ENVIRONMENT

In distributed firewall environments, it's possible to quickly visually perceive the status of misuse policies, and this visualization can be leveraged for detecting, comparing, and improving these policies. Within the context of distributed firewall settings, we propose a novel view that breaks down the distribution of misuse policies by each octet of the Source IP, offering a more granular perspective. By utilizing the proposed tool, one can ascertain the distribution of misuse policies in a distributed firewall environment, recognizing the need for policy refinement and improvement. It's feasible to identify which firewall has a high prevalence of misuse policies and, if necessary, use this information to persuade the relevant task managers of the need for policy enhancement. This tool proves especially beneficial when tasks are divided among multiple managers in large-scale settings, allowing for its application based on specific needs.

C. IDENTIFICATION AND IMPROVEMENT OF UNNECESSARILY PERMITTED SERVICE POLICIES

Information security operators must periodically check services for vulnerabilities and take measures to eliminate them. Understanding the correlations of misuse policies distributed across multiple firewalls and proceeding with policy improvement is not straightforward. In such cases, using the proposed tool can be beneficial. By utilizing this tool, one can assess the necessity of the allow policies applied across various firewalls and optimize these policies. This process aids in addressing potential vulnerabilities, thereby enhancing the overall security posture.

D. DECISION SUPPORT

The need for checking misuse policies can be swiftly addressed and effectively reviewed through the presentation of detailed visualization results. Firewall operators can utilize this tool to grasp the distribution status of firewall misuse policies, determining whether there's a need for policy improvement. Moreover, the tool can be employed to emphasize the necessity for policy enhancement, persuade decision-makers, and support various decision-making processes. If there's an excessive number of misuse policies, the tool allows for an intuitive check on which firewall has a significant distribution of these policies. Thus, with just one visualization, it can serve as a decision-making tool to determine the need for policy improvement efforts.

E. UTILIZATION AS A SECURITY AUDIT TOOL

If excessively permissive firewall policies that are not managed are applied, attackers can potentially gain access to internal systems. Misuse policies in firewalls can lead to vulnerabilities in service security, hence they need to be meticulously managed. By using the mentioned tool, one can quickly grasp the distribution of misuse policies at a glance. Therefore, it can serve as a security audit tool to analyze firewall policies and identify vulnerabilities.

V. EVALUATION

In this paper, we introduced and discussed the 64 visualization results of HSViz-II, which effectively displays the

TABLE 18. HSViz II performance test result: anomaly detection and visualization (1) shadowing, (2) spuriousness.

FW	Lines(c)	Allow(c)	Deny(c)	Parsing(s)	Shadowing(s)						Spuriousness(s)					
					count	A	A-B	B-C	C-D	D	count	A	A-B	B-C	C-D	D
FW1[U]	1565	1034	531	8.26	0	18.97	1.60	1.58	0.63	0.72	34.00	25.49	3.07	2.56	1.43	0.97
FW1[D]	244	240	3		42	25.28	4.96	4.55	1.63	1.55	12.00	26.70	4.02	2.17	1.47	1.17
BOTH	1809	1274	534		42	23.94	4.90	3.95	2.14	1.28	46.00	26.94	5.21	3.18	2.84	1.38
FW2[U]	1899	399	1500	12.34	0	43.49	1.37	1.24	1.98	0.27	81.00	76.16	2.94	3.33	3.28	3.37
FW2[D]	59	54	5		54	46.29	2.31	1.09	2.28	0.51	29.00	60.75	2.19	0.97	1.55	0.49
BOTH	1958	453	1505		54	47.22	2.51	1.16	2.35	0.56	110.00	78.53	5.29	4.16	4.80	3.61
FW3[U]	1644	387	1257	8.96	0	25.15	1.50	1.42	1.55	1.55	43.00	25.79	1.64	1.56	1.43	1.54
FW3[D]	132	125	7		39	26.44	3.83	2.21	1.95	2.11	20.00	25.65	1.92	2.35	1.99	1.99
BOTH	1776	512	1264		39	30.21	7.07	3.77	3.55	3.28	63.00	29.79	2.59	3.31	3.16	3.25
FW4[U]	92	87	5	5.99	0	**1.22**	**0.15**	0.16	0.20	0.19	29.00	1.66	0.15	0.11	0.13	0.13
FW4[D]	38	34	4		11	2.03	1.16	0.19	0.23	0.18	30.00	2.53	1.13	0.19	0.19	0.21
BOTH	130	121	9		11	2.30	0.23	0.21	0.20	0.22	59.00	2.80	0.28	0.26	0.25	0.20

TABLE 19. HSViz II performance test result: anomaly detection and visualization (3) redundancy, (4) correlation.

FW	Lines(c)	Allow(c)	Deny(c)	Parsing(s)	Redundancy(s)						Correlation(s)					
					count	A	A-B	B-C	C-D	D	count	A	A-B	B-C	C-D	D
FW1[U]	1565	1034	531	8.26	7	5.65	1.44	1.51	1.47	1.52	31.00	21.18	3.02	2.88	2.51	2.03
FW1[D]	244	240	3		0	3.71	3.78	2.24	2.24	2.18	8.00	21.35	3.93	2.15	2.07	2.14
BOTH	1809	1274	534		7	6.42	2.15	3.59	3.27	3.27	40.00	26.58	2.55	3.35	3.32	3.04
FW2[U]	1899	399	1500	12.34	10	5.29	2.11	1.11	1.16	0.89	43.00	73.42	3.28	3.32	3.32	3.23
FW2[D]	59	54	5		0	4.27	1.31	1.39	1.66	0.72	15.00	61.11	2.20	0.82	1.11	0.61
BOTH	1958	453	1505		10	5.42	2.18	1.66	1.29	0.92	58.00	79.54	3.24	3.61	3.60	3.64
FW3[U]	1644	387	1257	8.96	6	4.01	2.11	1.24	1.42	0.85	43.00	19.78	1.39	1.44	1.35	1.49
FW3[D]	132	125	7		0	2.94	1.40	1.49	1.07	0.75	0.00	16.67	0.06	0.09	0.06	0.06
BOTH	1776	512	1264		6	5.53	1.39	1.31	1.28	0.94	43.00	20.71	1.19	1.48	1.54	1.46
FW4[U]	92	87	5	5.99	3	1.50	0.81	0.83	0.43	0.46	24.00	1.16	0.12	0.11	0.11	0.12
FW4[D]	38	34	4		0	1.15	0.67	0.48	0.61	0.46	30.00	2.05	1.12	0.19	0.18	0.19
BOTH	130	121	9		3	1.90	0.70	0.58	0.52	0.50	54.00	1.24	0.14	0.13	0.14	0.14

distribution of policy misuse in distributed firewalls, and delved into the design process. In this chapter, we aim to validate the efficiency of the proposed tool by testing its performance using actual firewall policy data.

A. DATASET AND TEST ENVIRONMENT

The development environment was built on Python 3.7. For database management, we utilized MySQL along with the pymysql library. For the GUI implementation, PyQt5 was employed. Additionally, for the heatmap implementation, we made use of the matplotlib and seaborn libraries. Tests were conducted on actual operational firewall policies. Considering the Upstream firewall and Downstream firewall as one set, a total of 4 sets, or 8 firewall policies, were tested. The test environment was set up on a laptop equipped with a MAC OS M1 10 Core, 16G RAM, and a 512GB SSD.

B. PERFORMANCE

As indicated in the Table 18 and 19, four sets of upstream and downstream firewall policies, totaling 8 firewall policies, were used for performance measurement of HSViz-II. The number of ALLOW and DENY for each firewall policy was noted. The parsing speed was measured, and the number of misuse cases derived based on the four types of misuse cases for each firewall, as well as the visualization speed for each octet, were checked. "Count" was denoted as (c) and "second" as (s). The total line count and the allow/deny policies of each firewall policy were noted, along with the sum of the two firewall policies. The parsing speed of the two firewall policies was examined, and the extracted values of misuse policies were noted. Additionally, the time taken to visualize misuse policies by IP Octet was measured. Firewalls with a minimum of 38 lines to a maximum of 1958 lines were tested, and the parsing speed was recorded to take approximately 5.99 seconds to 12.34 seconds. As for the visualization speed, Shadowing took a maximum of 47.21 seconds, Spuriousness a maximum of 78.52 seconds, Redundancy a maximum of 6.42 seconds, and Correlation a maximum of 79.54 seconds.

VI. CONCLUSION AND FUTURE WORK

A. CONCLUSION

In this paper, we introduce HSViz-II, a newly designed tool for analyzing and comprehensively viewing the distribution of misuse policies in a distributed firewall environment. This novel visualization approach, not presented in previous studies, analyzes Upstream, Downstream, and Both policies and displays the distribution of anomalies based on the octets of the Source IP. For the four types of anomaly cases, it provides a total of 60 views, breaking down the Source IP into five octet layers. Through this tool, firewall operators can check the distribution of policies and anomalies by octet layer of the Source IP. The significance of this lies in the ability to view the distribution of misuse policies across distributed firewall policies from various perspectives, depending on the octet layer of the Source IP.

B. FUTURE WORK

In the future, beyond the visualization methods presented in HSViz-II, various visualization techniques can be devised to simplify complex firewall policies and assist operators in intuitively understanding them. There's potential for research into allowing users to select firewall policies for visualization based on their needs, or visually storytelling the changes in firewall policies over time. Additionally, research that effectively visualizes firewall policies in real-time is also necessary. As companies' network infrastructures become standardized around SDN-based and cloud-based network environments and are operated over the long term, efforts to address firewall policy misuse issues, such as visualization solutions, will continue to be a topic of ongoing research.

REFERENCES

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

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

[3] S. P. Morrissey and G. Grinstein, "Visualizing firewall configurations using created voids," in *Proc. 6th Int. Workshop Vis. Cyber Secur.*, Oct. 2009, pp. 75–79.

[4] M. Q. Ali, E. Al-Shaer, and T. Samak, "Firewall policy reconnaissance: Techniques and analysis," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 2, pp. 296–308, Feb. 2014.

[5] 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.

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

[7] 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.

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

[9] 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.

[10] 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.

[11] C.-S. Chao and S. J. Yang, "A bit vector-based diagnosis mechanism for firewall rule anomalies in IPv6 networking environment," *J. Internet Technol.*, vol. 22, no. 4, pp. 867–876, 2021.

[12] T. Kim, T. Kwon, J. Lee, and J. Song, "F/wvis: Hierarchical visual approach for effective optimization of firewall policy," *IEEE Access*, vol. 9, pp. 105989–106004, 2021.

[13] H. Hamed and E. Al-Shaer, "Taxonomy of conflicts in network security policies," *IEEE Commun. Mag.*, vol. 44, no. 3, pp. 134–141, Mar. 2006.

[14] C.-S. Chao and S. J.-H. Yang, "A novel three-tiered visualization approach for firewall rule validation," *J. Vis. Lang. Comput.*, vol. 22, no. 6, pp. 401–414, Dec. 2011.

[15] S. Pisharody, J. Natarajan, A. Chowdhary, A. Alshalan, and D. Huang, "Brew: A security policy analysis framework for distributed SDN-based cloud environments," *IEEE Trans. Dependable Secure Comput.*, vol. 16, no. 6, pp. 1011–1025, Nov. 2019.

HYUNJUNG LEE received the B.S. degree in computer engineering from Seoul Women's University, Seoul, in 2005, and the M.S. degrees in cybersecurity from Sungkyunkwan University, Seoul, in 2009. She is currently pursuing the Ph.D. degree with the School of Cybersecurity, Korea University, under the supervision of H. K. Kim.

Since 2011, she has been a Network and Security System Operator with KOSCOM, which developing and operating the core IT systems in Korean capital market and financial investment industry. Before joining KOSCOM, she was a Security System Operator with NCSOFT, from 2008 to 2010. She has performed penetration testing in various industries, when she was with A3 Security and SK Infosec, from 2005 to 2007. She has authored one security book and has translated two security books. Her research interests include network security, vulnerability analysis, data analysis, and visualization.

SURYEON LEE is currently pursuing the B.S. degree in computer engineering with Seoul Women's University, Seoul.

She has developed a web-based shooting game site and a location-sharing platform for students. She also performed a project to analyze media and SNS responses using text mining techniques. Her research interests include data analysis, visualization, and data privacy.

KYOUNGGON KIM (Member, IEEE) received the B.S. degree in computer science from Soongsil University, in 2008, and the M.S. and Ph.D. degrees in information security from Korea University, in 2015 and 2020, respectively. He is currently an Assistant Professor with the Department of Forensic Sciences, Naif Arab University for Security and Sciences (NAUSS). He has performed penetration testing for over 130 clients in various industries, when he was with Deloitte, PwC, and boutique consulting firms during over 15 years. He has authored a book on internet hacking and security and has translated numerous security books. His research interests include cybercrime and network forensics, vulnerability analysis, smart city security, and CPS and IoT security. He received a sixth place at DefCon CTF, in 2007, and a first prize at the First Hacking Defense Contest hosted by the Korea Information Security Agency.

HUY KANG KIM (Member, IEEE) received the B.S. degree in industrial management, the M.S. degree in industrial engineering, and the Ph.D. degree in industrial and systems engineering from the Korea Advanced Institute of Science and Technology (KAIST), in 1998, 2000, and 2009, respectively. He is a Professor with the School of Cybersecurity, Korea University. In 1999, he has founded A3 Security Consulting, the first information security consulting company in South Korea. Before joining Korea University, he was a Technical Director (TD) and the Head of the Information Security Department, NCSOFT, from 2004 to 2010, one of the most famous MMORPG companies in the world. His current research is focused on solving many security problems in online games based on the user behavior analysis.