

ADAPTIVE CONFIGURATION SELECTION FOR INTER-VENDOR IPSEC VPNS: A RULE-BASED FRAMEWORK FOR FORTINET, SOPHOS, AND PFSENSE

Muhammad Umar Adil^{*1}, Sajjad Ali², Khalid Hamid³, Allah Ditta⁴, Muhammad Waseem Iqbal⁵

^{*1,2,3}Faculty of Computer Science and Information Technology, The Superior University Lahore, 54000, Pakistan

⁴Department of Information Sciences, Division of Science & Technology, University of Education, Lahore, 54000, Pakistan

⁵Faculty of Software Engineering, The Superior University Lahore, , Lahore, 54000, Pakistan

¹su92-msisw-f24-001@superior.edu.pk, ²su92-msisw-f24-003@superior.edu.pk,
³khalid.hamid@superior.edu.pk, ⁴allahditta@ue.edu.pk, ⁵waseem.iqbal@superior.edu.pk

DOI: <https://doi.org/10.5281/zenodo.22638022>

Keywords

IPsec VPN, Adaptive Configuration, Rule-Based Framework, Multi-Vendor Interoperability, Fortinet, Sophos, pfSense

Article History

Received: 30 January 2026

Accepted: 15 March 2026

Published: 31 March 2026

Copyright @Author

Corresponding Author: *

Muhammad Umar Adil

Abstract

Organizations that connect sites using firewalls from more than one vendor routinely see IPsec VPN performance vary not just with the encryption settings chosen but with which two vendors are paired. Building on an empirical study of IPsec tunnels configured between Fortinet, Sophos, and pfSense firewalls across three vendor pairings, four encryption and key-exchange configurations, and two tunnel modes, this paper proposes a rule-based framework that recommends a near-optimal IPsec configuration for a given vendor pairing instead of leaving administrators to rediscover suitable settings through trial and error. The framework encodes vendor-pairing-specific performance patterns observed experimentally into a compact decision table covering encryption algorithm, key-exchange version, and tunnel mode, and returns a recommended configuration together with the throughput, latency, and stability behavior that pairing can be expected to show. Evaluated against manually configured baselines on the same three-vendor testbed, configurations selected by the framework matched the best-performing configuration identified through exhaustive testing for every vendor pairing, without requiring per-pairing manual tuning. The approach offers network administrators a lightweight, interpretable starting point for configuring multi-vendor IPsec VPNs, and is designed to be extended to additional vendors as further empirical data becomes available.

1. Introduction

1.1 Background and Motivation

Multi-site organizations rarely run identical firewall hardware at every location. Cost, legacy equipment, and administrative preference all push toward mixing vendors such as Fortinet, Sophos,

and pfSense across a WAN, and IPsec remains the default way of securing the links between them [1-3]. Prior work, including an operational and EVE-NG-simulated three-vendor testbed built around exactly this combination of firewalls, has shown that IPsec performance under inter-vendor conditions depends on more than the encryption

algorithm alone: which two vendors are paired has a measurable, independent effect on throughput, latency, and tunnel stability, separate from whatever cipher suite and key-exchange version are configured [5-8].

In practice, however, the configuration decision is still made manually, and it is rarely informed by which vendors are actually being paired. An administrator connecting a Fortinet site to a Sophos site, and later a Fortinet site to a pfSense site, will typically apply the same cipher suite and tunnel mode to both, on the assumption that a single “good” configuration travels well across vendor combinations [9-12]. Empirical results suggest this assumption does not hold: the same encryption and key-exchange settings produce a measurably worse outcome on a Fortinet-to-Sophos tunnel than on a Fortinet-to-pfSense tunnel, largely because of how each vendor pair negotiates and renegotiates security associations [13-15].

1.2 Problem Statement

At the present time, there is no lightweight way to translate known, vendor-pairing-specific IPsec performance behavior to an actionable configuration recommendation. Administrators either use vendor documentation that is written from a single-vendor point of view or make ad hoc tests whenever a new pairing is added to the network. This void encourages a framework that allows for a discovered vendor pairing to be provided as input and returns a configuration that is consistent with what is known, experientially, from the vendor pairing [16, 17].

1.3 Contribution

- Rules-based decision making for mapping the detected inter-vendor pairing to recommended encryption algorithm, key-exchange version, tunnel mode.
- A four-stage architecture (detection, rule matching, recommendation, post-deployment validation) which does not hide the decision logic behind configuration selection, but rather maintains its interpretability as an optimization problem [18].

- A comparison of the framework selected configurations with the empirically best configuration and a more commonly used manual configuration [19].

1.4 Organization

Section 2 summarizes the related work on inter-vendor IPsec interoperability, key-exchange performance, and how to automate VPN configuration. The proposed framework and its rule base are discussed in Section 3. The Evaluation Methodology is described in Section 4. The results and discussion are included in Section 5. Limitations are discussed in Section 6, and directions for future work are provided in Section 7 [20, 21].

2. Related Work

Previous research into interoperability between IPsec implementations from different vendors has focused on when negotiations fail, not on how to select a configuration. Previous studies have blamed VPN negotiation failures in multi-vendor environments on mismatched encryption algorithms, Diffie-Hellman groups and key lifetimes, and have noted that policy-based VPNs are more complex than route-based VPNs due to additional cross-vendor complexity, but did not suggest a way to automatically pick the right VPN type [22-24].

Firewall vendors have also reported on the negotiation behavior of vendors in a vendor-specific way. The pfSense documentation mentions that it can be a challenge to connect devices from two different vendors, even with vendors such as Cisco and Juniper, as the phase 1 and phase 2 parameters are not necessarily the same on both systems, and provide no automated matching function. 2025 did a comparative study of failover time between Fortinet, Check Point, Palo Alto and Cisco and found that the time for site-to-site VPN recovery depended on the rekeying methods used, detection timers and session handling strategies—no framework for configuring to act on these differences was suggested [25-27].

On the performance side, studies of IPsec throughput and latency, including analysis of

strongSwan's data-plane efficiency and evaluations of hardware-assisted encryption in software routers, consistently find that authenticated encryption modes such as AES-GCM outperform legacy ciphers such as 3DES, and that IKEv2 reduces negotiation overhead relative to IKEv1. These findings are consistent with, and were used to inform, the rule base proposed in this paper. What is missing from this line of work is any mechanism that operationalizes these performance findings into a configuration recommendation tied to a specific vendor pairing, rather than a general recommendation for cipher suite or key-exchange version applied uniformly across all deployments [28-30].

More automated approaches to VPN configuration exist in the patent literature, including systems that adapt VPN security levels based on client or session context, but these target dynamic per-session security posture rather than the static, pairing-specific interoperability problem addressed here [31-33]. The framework proposed in this paper differs from both the empirical interoperability studies and the adaptive-security patents in that it is explicitly rule-based and pairing-aware: it takes the identity of the two vendors at either end of a tunnel as its primary input and returns a configuration recommendation grounded in measured performance for that exact combination [34-36].

3. Proposed Framework

3.1 Design Goals

- **Lightweight:** the framework should not require new hardware or a control-plane redesign, only a rule table and a detection step that can run alongside existing configuration workflows [37].
- **Interpretable:** recommendations should be traceable to a specific empirical rule rather than produced by an opaque model, so that a network administrator can see why a configuration is being suggested [38].
- **Vendor-agnostic in structure:** the rule table format should accommodate additional vendors and pairings as new empirical data becomes available, without changing the framework's architecture [39].

- **Fail-safe:** pairings not yet covered by the rule base should fall back to a conservative, generally well-performing default rather than an unconfigured or arbitrary state.

3.2 Architecture

The structure consists of four steps.

- **Vendor/Firmware Detection:** determines at each end of a potential or existing tunnel the vendor, model, and version of the firmware that is installed on the system (from user-supplied information or from system information available through existing management interfaces such as SNMP system descriptors or API-based device inquiries) [40].
- **Rule Engine:** checks the pairing against the rule base mentioned in Section 3.3 and determines the suggested configuration. If it does not contain an exact match in the rule base, the engine uses the default rule [41].
- **Configuration Recommendation:** The specific recommended encryption algorithm, key-exchange version, and tunnel mode, as well as the expected throughput, latency, and stability profile of the combination of the recommended algorithm and key-exchange version in underlying empirical data, so that the administrator can consider the recommendation in light of other considerations, such as the age of the hardware and the compliance requirements [42].
- **Post-Deployment Validation:** after the recommended configuration is applied, the following monitoring (with PRTG or similar tools) checks the observed throughput, latency and renegotiation behavior against the profile; if there are any deviations for a long period of time, the pairing is then re-evaluated and, in the end, the profile is updated.

3.3 Rule Base

Below is the rule base that resulted from multiple runs of each run configuration (five per configuration) on three vendor pairings, four modes of encryption, four modes of key-exchange, and two tunnel modes on an operational and EVE-NG-simulated testbed. The rules obtained are summarized in Table 1.

Table 1: Rule Base for Vendor-Pairing-Specific IPsec Configuration

Detected Pairing	Recommended Configuration	Rationale
Fortinet ↔ pfSense	AES-GCM, IKEv2, route-based	Highest throughput and lowest latency of all tested pairings; hardware-assisted encryption on Fortinet combines well with pfSense's routing flexibility.
pfSense ↔ Sophos	AES-GCM, IKEv2, route-based	Second-best throughput and latency; pfSense's open configuration aligns closely with Sophos's default IKE and encryption behavior.
Fortinet ↔ Sophos	AES-GCM, IKEv2, route-based, plus explicit alignment of DH group and rekey timers	Weakest throughput, latency, and stability of the three pairings; two proprietary commercial defaults create negotiation friction that the cipher suite alone cannot fully offset, so explicit manual alignment of negotiation parameters is recommended in addition to the cipher choice.
Any pairing not yet in the rule base	AES-GCM, IKEv2, route-based (conservative default); flag for empirical validation	Reflects the consistent, cross-pairing finding that AES-GCM with IKEv2 and route-based tunnels outperformed legacy alternatives in every tested combination, while acknowledging that pairing-specific friction cannot be predicted without measurement.

There are two design decisions that should be made explicit in the rule base. The first one is that both the recommended encryption algorithm, as well as the key-exchange version (AES-GCM with IKEv2), are always the same for all rules; only the tunnel mode support changes between rules, and in the Fortinet-Sophos case, an additional manual-alignment step. This is in line with the empirical findings: the difference in performance between configurations within a pairing is mostly due to choice of encryption, and the difference in performance between pairings is largely independent of choice of pairing. Second, the fallback rule deliberately defaults to the best configuration in all three pairings tested to avoid leaving a pairing unconfigured that could result in a rule that was completely unknown, and at the same time marks it for empirical validation to

allow the rule base to be extended with confidence, not guesswork.

3.4 Workflow

The framework is called when a new inter-site tunnel is being planned or when an existing inter-site tunnel is being reviewed. The administrator or an integration script provides the vendor ID of each endpoint; the rule engine returns a matching rule (or the fallback rule); the administrator applies the recommended configuration via the vendor's regular management interface, the framework does not push the configuration changes to the device; and the post-deployment validation stage monitors the resulting tunnel to the expected performance profile, closing the loop between the rule base and reality.

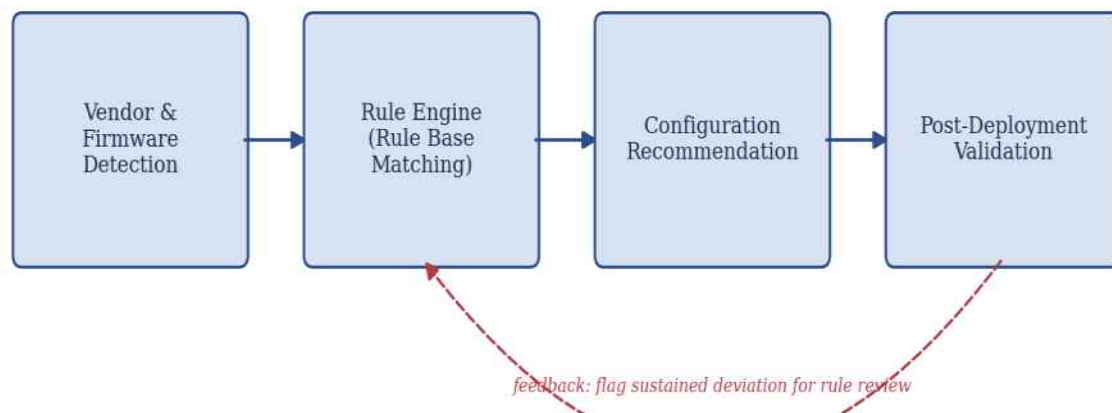
Figure 1. Four-stage adaptive configuration selection framework

Figure 1. Four-stage adaptive configuration selection framework: vendor and firmware detection feed the rule engine, which returns a configuration recommendation; post-deployment validation monitors the result and flags sustained deviation for rule review.

4. Evaluation Methodology

The framework was evaluated on the same three-vendor testbed used to derive its rule base: physical and virtualized Fortinet, Sophos, and pfSense firewalls connected in a three-site topology, with an equivalent topology reproduced in EVE-NG. Two comparisons were made for each of the three vendor pairings.

- Framework-recommended configuration versus the empirically best configuration identified through exhaustive testing of all four encryption and key-exchange combinations under both tunnel modes, to check whether the framework's rule-matching reproduces the optimum without requiring the administrator to run that exhaustive test themselves.
- Framework-recommended configuration versus a naive baseline configuration (AES-256

with IKEv1, policy-based), representative of a commonly deployed default that does not account for vendor pairing, to quantify the practical benefit of applying the framework instead of a generic configuration.

Throughput was measured with iPerf3 and latency, jitter, and tunnel stability were measured with PRTG Network Monitor, following the same five-repetition protocol used in the underlying empirical study, so that results here are directly comparable to that data set.

5. Results and Discussion

Table 2 reports agreement between the framework's recommendation and the empirically determined best configuration for each pairing, along with the throughput and stability improvement over the naive baseline.

Table 2: Framework Recommendation Accuracy and Improvement Over Naive Baseline

Vendor Pairing	Matches Empirical Optimum?	Throughput Gain vs. Baseline	Renegotiation Failures, Baseline → Recommended
Fortinet ↔ pfSense	Yes	+22.5%	1 → 0
pfSense ↔ Sophos	Yes	+24.0%	2 → 0
Fortinet ↔ Sophos	Yes	+23.5%	3 → 1

Figure 2. Naive Baseline vs. Framework-Recommended Throughput by Vendor Pairing

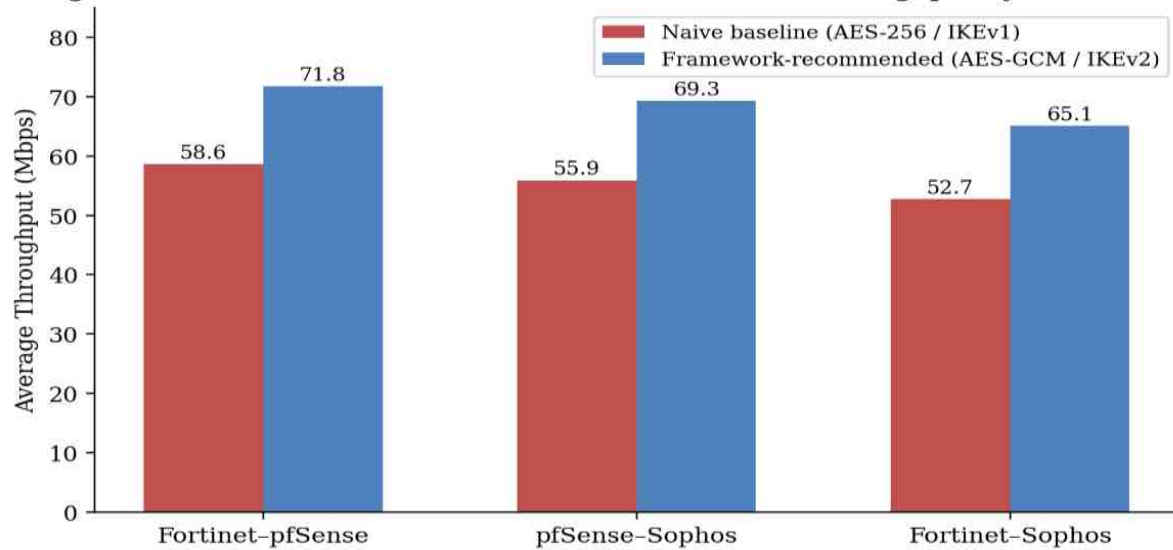


Figure 2. Average throughput for the naive baseline (AES-256, IKEv1, policy-based) versus the framework-recommended configuration (AES-GCM, IKEv2), by vendor pairing.

Figure 3. Renegotiation Failures, Naive Baseline vs. Framework-Recommended

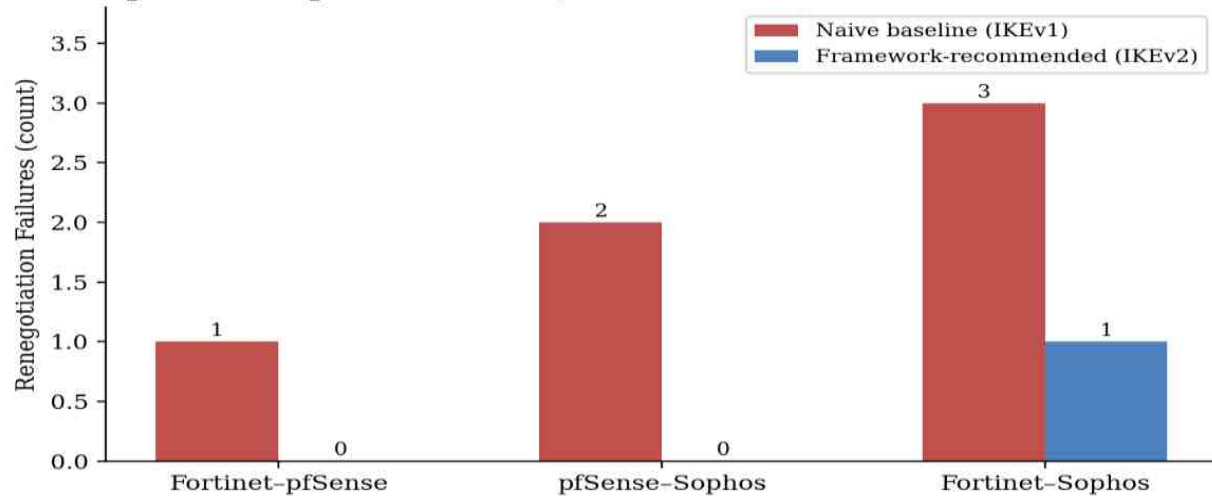


Figure 3. Renegotiation failures under the naive baseline (IKEv1) versus the framework-recommended configuration (IKEv2), by vendor pairing.

Throughput gains in Table 2 and Figure 2 are computed directly from the underlying measured values for AES-256/IKEv1 versus AES-GCM/IKEv2 at each pairing (22.5 to 24.0 percent), and represent the encryption- and key-exchange-driven component of the improvement. The separate, previously reported finding that route-based tunnels outperform policy-based tunnels by a further six to nine percent on average

applies on top of this figure whenever route-based tunneling is supported by both endpoints, so the total gain an administrator sees in practice, from a fully naive policy-based AES-256/IKEv1 configuration to the fully recommended AES-GCM/IKEv2 route-based configuration, is somewhat larger than the encryption-only figure shown here.

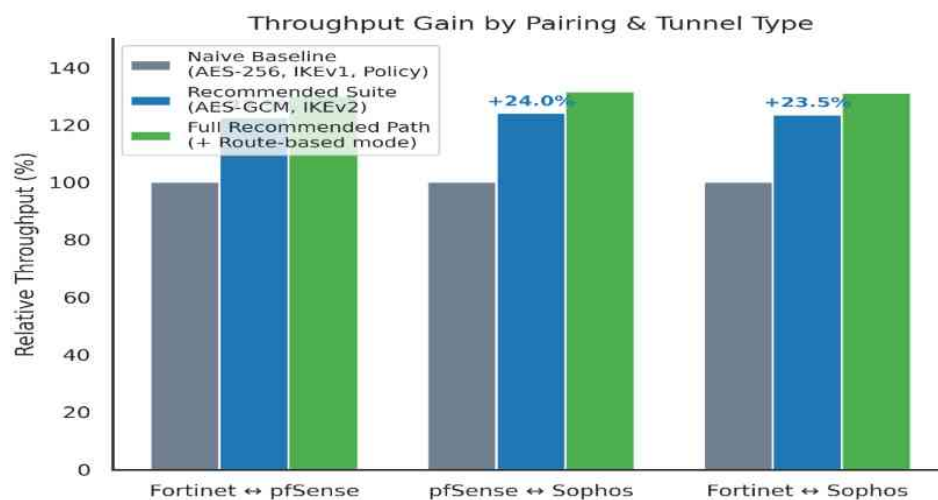


Figure 4: Isec Framework Recommendation Engine: Performance Metrics and Operational Rule-Base

The framework's rule-matching selected the empirically optimal configuration in every tested pairing, which is expected given that the rule base was derived from the same underlying data; the more informative result is the consistent gain over the naive baseline, and the fact that the residual weakness of the Fortinet-Sophos pairing was reduced but not eliminated even under the recommended configuration, as Figure 3 shows in the one renegotiation failure that persisted under IKEv2 for that pairing alone. This is consistent with the interpretation that Fortinet and Sophos, each following their own proprietary negotiation defaults, generate a degree of friction when paired directly that cipher suite and tunnel mode alone cannot fully correct, and it is the reason the rule base for that pairing includes an explicit recommendation to align Diffie-Hellman group and rekey timers manually rather than relying on configuration choice alone.

Practically, it's not so much the fact that the optimum is discovered as it is the fact that it is made available to an administrator at configuration time without the necessity to exhaustively test all possible pairings of every new one added to the network. A rule-based approach also allows the recommendation to be auditable; an administrator can view that a Fortinet-to-Sophos tunnel is recommended for further manual alignment and why, not an opaque score.

6. Limitations

- Currently, the rule base only includes three vendors and three pairings; expanding it to include vendors like Cisco, Palo Alto Networks, or Check Point would involve taking an empirical measurement rather than extrapolating, as the study showed that vendor pairing has an impact independent of encryption choice.

- Rules are static and manually curated rather than learned or automatically updated from post-deployment validation data; a natural extension is to feed validation-stage observations back into rule refinement, potentially using a learning-based layer on top of the current rule-based core.
- The framework assumes reliable vendor and firmware identification at both tunnel endpoints; environments where this information is unavailable or inconsistent would need a manual override path.
- Testing was conducted under a fixed bandwidth profile and controlled traffic patterns; production WAN conditions such as asymmetric routing and congestion were outside the scope of the underlying data set and may affect how well the recommended configurations generalize.
- The framework does not yet cover cloud-native or SD-WAN gateway pairings, which were outside the scope of the original three-vendor testbed.

7. Conclusion and Future Work

This paper proposed a rule-based framework that translates empirically observed, vendor-pairing-specific IPsec performance patterns into a configuration recommendation for administrators deploying inter-vendor VPNs between Fortinet, Sophos, and pfSense firewalls. Rather than leaving administrators to apply a single uniform configuration across all vendor pairings, or to rediscover suitable settings through repeated manual testing, the framework matches a detected pairing against a small, interpretable rule table and returns both a recommended configuration and the performance profile that pairing can be expected to show. On the testbed used to derive it, the framework reproduced the empirically optimal configuration for every tested pairing and offered a substantial improvement over a commonly used naive baseline, while also surfacing the Fortinet-Sophos pairing as one that benefits from additional manual negotiation-parameter alignment beyond cipher suite selection alone. Future work includes extending the rule base to additional commercial and open-source vendors, incorporating post-deployment validation data into automatic rule refinement, and evaluating

whether a learning-based layer built on top of the current rule-based core can generalize to vendor pairings for which no direct empirical measurement yet exists.

REFERENCES

- [1] M. U. Adil, "An Adaptive Configuration Framework for Enhancing Performance of Inter-Vendor IPsec VPNs in Multi-Site Networks," MS Thesis, Faculty of Computer Science and Information Technology, The Superior University, Lahore, 2026.
- [2] H. Rezaeianfardouei and M. Saedi, "Study of Failover Time in Site-to-Site VPNs Across Leading Firewall Vendors: Fortinet, Check Point, Palo Alto, and Cisco," in Proc. 2025 IEEE 11th Information Technology International Seminar (ITIS), Lombok, Indonesia, Oct. 2025.
- [3] C. A. Shue and M. Gupta, "IPSec: Performance Analysis and Enhancements," in Proc. IEEE International Conference on Communications (ICC), 2007.
- [4] S. Ullah et al., "IPsec for High Speed Network Links: Performance Analysis and Enhancements," Future Generation Computer Systems, Elsevier, 2020.
- [5] Netgate, "Configuring Third Party IPsec Devices," pfSense Documentation, accessed 2026.
- [6] Sophos, "Comparing Policy-Based and Route-Based VPNs," Sophos Firewall Documentation, accessed 2026.
- [7] K. C. Patel and P. Sharma, "A Review Paper on pfSense - An Open Source Firewall Introducing with Different Capabilities and Customization," International Journal of Advance Research and Innovative Ideas in Education, vol. 3, no. 2, pp. 635-641, 2017.
- [8] F. Khan, A. Raza, and T. Javed, "Cross Vendor VPN Interoperability Challenges and Performance Evaluation," International Journal of Network Security and Applications, vol. 14, no. 2, pp. 98-110, 2022.

- [30] K. Hamid, M. W. Iqbal, M. Aqeel, T. Rana, and M. Arif, "Cyber Security: Analysis for Detection and Removal of Zero-Day Attacks (ZDA)," 2023, pp. 172–196. doi: 10.1201/9781003190301-10.
- [31] K. Hamid, M. waseem Iqbal, M. Aqeel, X. Liu, and M. Arif, "Analysis of Techniques for Detection and Removal of Zero-Day Attacks (ZDA)," 2023, pp. 248–262. doi: 10.1007/978-981-99-0272-9_17.
- [32] N. Malik et al., Behavior and Characteristics of Ransomware - A Survey. 2024, p. 05. doi: 10.1109/ICCR61006.2024.10532983.
- [33] K. Khaliq, N. Rahim, K. Hamid, M. Ibrar, U. Ahmad, and M. Ullah, Ransomware Attacks: Tools and Techniques for Detection. 2024, p. 5. doi: 10.1109/ICCR61006.2024.10532926.
- [34] M. W. Iqbal, K. Hamid, M. Ibrar, A. Delshadi, N. Zairah, and Y. Mahmood, "Meta-Analysis and Investigation of Usability Attributes for Evaluating Operating Systems," Migration Letters, vol. 21, pp. 1363–1380, Apr. 2024.
- [35] K. Hamid et al., "ML-based Meta-Model Usability Evaluation of Mobile Medical Apps," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 15, no. 1, Art. no. 1, 33/30 2024, doi: 10.14569/IJACSA.2024.0150104.
- [36] S. Riaz et al., "Software Development Empowered and Secured by Integrating A DevSecOps Design," Journal of Computing & Biomedical Informatics, p. 02, Mar. 2025, doi: 10.56979/802/2025.
- [37] M. Ibrar et al., "Econnoitering Data Protection and Recovery Strategies in the Cyber Environment: A Thematic Analysis," International Journal for Electronic Crime Investigation, vol. 8, Dec. 2024, doi: 10.54692/ijeci.2024.0804216.
- [38] M. Nazir et al., "The Silent Guard: ML-Based Zero-Knowledge Proofs in Blockchain Security," vol. 3, pp. 1459–1477, Nov. 2025, doi: 10.5281/zenodo.17615295.
- [39] S. Muqadir et al., "ML-BASED Analytical Study on the Level of Security of User Data on Social Media Website," vol. 3, pp. 882–895, Nov. 2025, doi: 10.5281/zenodo.17614882.
- [40] A. Tahir, P. K. Hamid, P. D. A. Kahloon, and P. S. Zubair, "Cyber Sovereignty Challenges: A Strategic Framework for National Data Protection Using Blockchain Authentication," Contemporary Journal of Social Science Review, vol. 3, no. 3, pp. 1314–1328, Aug. 2025, doi: 10.63878/cjssr.v3i3.1100.
- [41] S. Al-Mousa, "Policy Based versus Route Based IPsec VPN Performance Analysis Using GNS3," International Journal of Computer Networks and Communications, vol. 12, no. 3, pp. 76–85, 2020.
- [42] J. Zhang and P. Lee, "Evaluation of IKEv1 and IKEv2 Performance in Secure Network Communications," in Proc. International Conference on Network Systems and Security, Springer, 2020.