INTERNATIONAL JOURNAL OF ADVANCED RESEARCH
ISSN: 3106-7980: 3106-7972
Volume 3, Issue 2, 2026

ANALYTICAL STUDY FOR DETECTING VPN-BASED FIREWALL BYPASS IN ENTERPRISE NETWORKS: A TRAFFIC FLOW INVESTIGATION USING UNIFI UDM PRO

Malik Waqar Ali[*1], Khalid Hamid[2], Muhammad Waseem Iqbal[3]

[*1]*Department of Information Security, Superior University Lahore, 54000, Pakistan*
[2,3]*Department of Computer Science and IT, Superior University Lahore, 54000, Pakistan*

[*1]su92-msisw-s26-001@superior.edu.pk, [*1]waqaralibcs076@gmail.com, [2]khalid6140@gmail.com, [3]waseem.iqbal@superior.edu.pk

DOI: https://doi.org/10.5281/zenodo.21544738

Keywords
Firewall, VPN, visibility, Tunnelling, UDM Pro, TLS, Signature matching

Article History
Received: 28 April 2026
Accepted: 15 June 2026
Published: 30 June 2026

Corresponding Author: *
Malik Waqar Ali

Abstract
A firewall is a device that we use to protect our network and devices from unauthorized access. It controls user access to different platforms (Websites/Applications) by applying security policies. On the other hands VPN (Virtual Private Networks) are commonly used to create secure transmission through encrypted tunnels. However, the use of VPNs introduces significant challenges for network administrators and firewall systems. When a client activates a Public/Third-party VPN connection, network traffic is routed through an encrypted tunnel. And it will limit the visibility of the firewall and bypass the firewall policies. As a result, traditional firewall mechanisms may fail to enforce security policies effectively. This study analyses the limitations of firewall visibility under VPN traffic tunnelling by Third-Party VPNs. A real-world experimental setup is implemented using the Ubiquiti UniFi Dream Machine Pro, where security policies are configured, and system behaviour is observed before and after VPN activation. Based on the findings, we propose a layered base detection approach that identifies firewall blind spots and detects VPN traffic. An allow-list is used to make sure that trusted business VPN traffic is not identified as unauthorized VPN traffic.

1. Introduction

In today's digital environment, as technology grows continuously, it also increases the risks related to network security. With the rise in internet usage, cyber threats are becoming more serious and frequent. To protect networks, firewalls are widely used around the world [21]. They help control network traffic based on predefined security rules such as Access Control Lists (ACL), application filtering, and traffic inspection. Every organization needs to ensure the confidentiality, integrity, and availability of its data [1][20].

Virtual Private Networks (VPNs) are commonly used to secure user data by encrypting communication and sending it through a protected tunnel across networks. As the need for secure communication increases, the use of VPNs has also grown rapidly. National Institute of Standards and Technology guidance states that in remote access or online transmission the encryption is very important to secure communication [2].

As VPN use increases day by day for smooth and secure access to remote site there is an increase in challenges for IT Administrators. Because the traffic of a VPN device passes through a secure tunnel it decreases firewall visibility, that cause lake of inspection and monitoring. As firewall works on deep packet inspection DPI and application-level filtering, the use of VPN limited the inspection of these techniques [22]

[23]. And this tunnel makes traffic pass without inspection. Also, recent research shows that encrypted VPN traffic decreases the effectiveness of network firewalls and makes it harder for security systems to detect security threats and enforce policies [3].
In enterprises, the Network administrator have to apply many policies on the network, like website blocking, Application Blocking, Countries restriction & domain blocking, for smooth and secure network traffic flow. But the use of VPN creates a bypass of those policies and creates security challenges for the IT Team. Activation of VPN in client devices hides their activities and creates blind spots for traditional firewall/security systems [4]. And these types of client activity create serious concerns for an IT Administrator of an enterprise environment to apply his secure policies.
As shown in Figure 1.

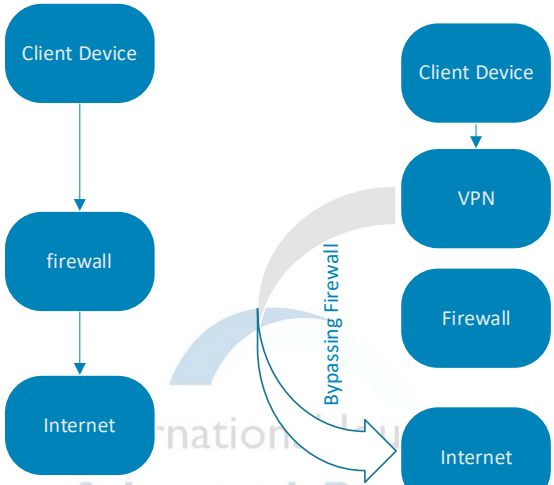

Figure 1: VPN Bypassing Firewall

After monitoring these concerns of network security, the Next-Generation firewalls and zero trust architectures have introduced modern security solutions [2].
The modern techniques working on [24]
1. SSL Inspection
2. Activity Monitoring
3. Identity-based access
to overcome the security bypass issues created by VPNs. However, these overcome the visibility limit, but in a real-world environment, these solutions can increase system complexity. Also, effect performance and increase privacy breaches [5][19].
This paper is motivated by a real operational scenario: an organization using a UDM Pro to restrict access to specific categories of websites under a policy named "Blocking Social Apps," where users have begun using VPN clients to circumvent these restrictions. Rather than treating this as a purely theoretical problem, we captured live firewall traffic logs from two client devices on the network, each before and after activating a different VPN service, to directly observe what changes and what does not. We then used those observations to build a detection framework, one that also explicitly avoids flagging the organization's own legitimate VPN traffic.
This study gives a five-layer detection framework combining.
1. Protocol signature matching
2. TLS fingerprinting
3. IP reputation analysis
4. flow-based behavioural scoring
5. Client attribution
with an explicit allow-list mechanism to avoid flagging the organization's own legitimate business VPN traffic, such as site-to-site links, remote server administration, and cloud infrastructure access.

2. Literature Review

With the expansion of Digital infrastructure and the increase of cyber-attack the network security plays an important role in the digital world. Most daily life and business activities run over the internet and to make these activities secure, there is a need for an internet security system [25][26]. To make our activities and network infrastructure secure, there is a need for a firewall. Firewalls play an important role between the user and attacker. It enforces different rules:

1. Access control lists
2. Application filtering
3. Traffic inspection
4. Deep packet inspection DPI
5. Web/Application level blocking

for any organization that needs to protect the confidentiality, integrity and availability of its data [1].

These tasks help to protect internal networks from external networks. A firewall plays the role of a barrier between the network and cyber-attackers. In enterprises/modern structured companies, a firewall is considered the first line of defence system. There are different firewalls available, but all have different abilities to monitor and make infrastructure secure by inspecting traffic [27][28].

On the other hand, a VPN works for secure transmission of data through different platforms (Site-to-Site access, Cloud access, Remote access) and makes our communications secure. It uses a secure VPN tunnel for transmission of data over the internet to make it safe from unauthorized access of third person interference. According to the National Institute of Standards and Technology NIST [2], a VPN is essential for working in a remote environment and making secure communications because VPN use a secure tunnel for transmission. Its transfer encrypt data through a tunnel so that an unauthorized person can't access to data. This helps to maintain the CIA Triad of security. Make the system confidential and secure its integrity.

As the VPN makes our system transmission secure the Third-Party/Open-Source VPNs also is a big challenge for security administrators to detect them. Firewalls in enterprises are used to control access by applying different policies, but the use of Third-Party/Open-Source VPNs in connected devices is making firewalls blind to detect traffic flow. Clients will use them to bypass the firewall policies to access those blocked sites/applications. Cisco also reported that the use of encrypted traffic tunnels significantly reduces the ability of security tools to inspect and analyse the traffic properly [3]. As a result, older/traditional firewalls facing limitation will inspecting traffic to detect threats, monitor clients' activities and enforce security policies.

The VPN's encrypted traffic flow can bypass the firewall restriction/policy in an enterprise network and creates blind spots which reduce the firewall system's effectiveness and efficiency of monitoring traffic.

Modern Open-Source VPN also have adopted different traffic patterns to bypass the firewall restrictions. Some of them use a traffic tunnel over port 443 because the same port is used for HTTPS web browsing and the firewall can't block this port; therefore, only port-blocking rules can't distinguish VPN from a normal web request [9][18].

Some VPNs use obfuscation techniques to hide the VPN handshake and avoid detection by DPI deep packet inspection [7].

Much existing research focuses on detecting and classifying VPN traffic using machine learning and [7]. But there are only a few studies are focusing on improving the visibility limit of the firewall to

VPN traffic by combining multiple VPN traffic flow patterns.

To cover this gap, this research paper proposes a layer-based VPN detection framework that combines.

1. VPN Protocol Signatures
2. TLS Fingerprinting
3. IP reputation analysis
4. Network flow behavioural analysis
5. Client Contribution
6. Allow List

to ensure VPN detection.

By applying this layer-based approach, the research aims to identify the blind spots in traffic monitoring by firewall that create by Open-Source VPNs.

This literature shows that firewalls are used for secure networks, but the interference of Open-

Source VPNs make his traffic monitoring visibility limited [29]. This limitation creates challenges for firewalls to detect threats and enforce security policies on the network. As a result, researchers are focusing on intelligent solutions.

Table 1: Comparative view

Ref.	Focus	Detection signal used	Key limitation
[1]	IoT security and forensics	None (general)	Not VPN-specific
[2]	Zero Trust architecture	Identity-based access	Conceptual, not an implemented detector
[3]	Internet traffic trends	None (descriptive)	No detection method proposed
[4]	Encrypted traffic classification	Machine learning	Single signal, no allow-list
[5]	Deep learning for cybersecurity	Deep learning (general)	Broad survey, not VPN-specific
[6]	Network traffic classification	Deep learning	Single signal, no live validation
[7]	Encrypted traffic analysis	Machine learning	Single signal, no allow-list
[9]	Commercial VPN ecosystem	Port/protocol behaviour	Descriptive, no live detection framework

3. Methodology

We work on a real-time firewall and client for our experiment to detect traffic flow before and after Open-Source/Third-Party VPN activation in client devices [30][31].

3.1 Traffic Flow Analysis Before and After VPN Use

This study includes an experimental research approach to investigate the traffic flow before and after VPN activation in client devices, as a social media blocking policy is activated in the firewall. There, we detect the limitations of the firewall before and after VPN activation in client devices.

This research follows a comparative analysis of traffic flow through the network under two different scenarios:

1. Normal Internet Traffic (Without VPN)
2. Encrypted Internet Traffic (With VPN Enabled)

To find results, we get data from firewall logs before and after VPN activation over a certain time period; we finalize which difference occurs in both traffic flows before and after.

DNS behaviour: In normal browsing, a client generates a steady stream of DNS lookups to many different domains as it loads pages, ads, and embedded content. Once a VPN is active, DNS resolution is often handled by the VPN client itself or routed through the tunnel, so the local network may see a drop in DNS queries from that client — though, as shown in Section 4, this drop is not guaranteed and depends on whether the VPN client fully controls DNS resolution or leaves some of it exposed [12].

Destination diversity: Ordinary browsing produces connections to many different IP addresses and domains over the course of a session. A full-tunnel VPN collapses this into a single sustained connection to one external IP — the VPN server — for the duration of the session. A partial or per-app VPN, by contrast, may not produce this collapse at all.

Session duration and byte volume: VPN tunnels tend to be long-lived, often lasting as long as the user is connected, with a large share of the session's total byte volume concentrated in a small number of long-lived flows rather than spread evenly across many short ones.

TLS handshake fingerprint: Even though the content of a TLS connection is encrypted, the handshake that sets up the connection is not. The specific way a client constructs its TLS "ClientHello" message. which ciphers it offers, in what order, and which extensions it includes

forms a fingerprint, commonly captured as a JA3 hash, that differs between a normal browser and a VPN client's own TLS stack.

3.2 Proposed Method

The proposed method consists of implementing firewall security policies on Ubiquiti UniFi Dream Machine Pro (UDM Pro) and observing the traffic flow before and after VPN activation on a client device (PC/Mobile etc), there, we use a client PC.

We will follow different steps as [32][33]:
1. Configure firewall rules (blocking of Website/Social-Apps) in UDM Pro.
2. Connect 2 different client Laptop/PC with network of that firewall through Wifi/Network-cable.
3. Generate traffic before VPN activation for a specific time duration.
4. Save traffic logs from the firewall from monitoring.
5. Enable a Third-Party/Open-Source VPN on the client device (Laptop/PC) that we have connected to the network.
6. Try to open that website/Apps that we have blocked in firewall rules.
7. Observe firewall limitations to block those sites/apps.
8. Save logs for monitoring.
9. Compare both logs.
10. Identify firewall limitations and change of traffic flow of that client.

3.3 System Architecture

Architecture Description

This experiment will have:

- Internet Connection
- UniFi Dream Machine Pro (Firewall/Gateway)
- Client Workstation
- VPN Service Provider
- Monitoring and Logging Interface

3.4 Architectural Diagram

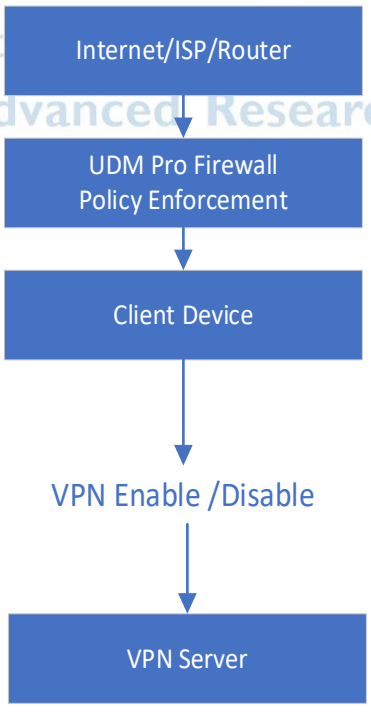

Figure 2: Architectural Diagram

3.4 Experimental Flowchart

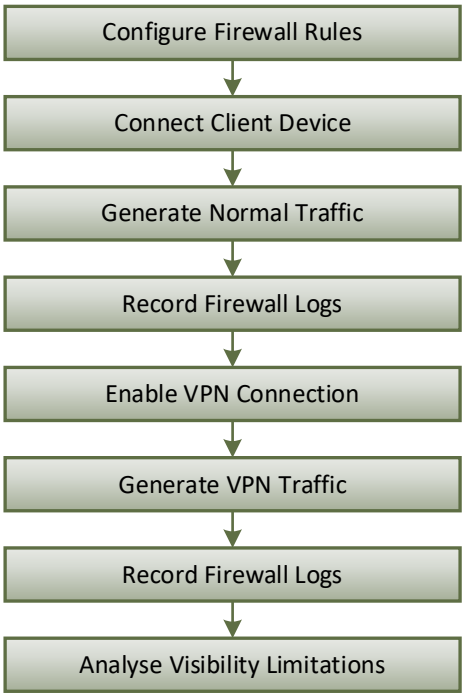

Figure 3:Experimental Flowchart

3.5 Data Collection

Data is collected from Firewall (UDM Pro) logs for a certain time, during which we have generated traffic for the experiment.

Data Sources

- UDM Pro Firewall Logs
- Traffic Statistics
- Application Detection Records
- Website Access Records
- VPN Connection Information

Collected Parameters

Table 2: Collected Parameters

Sr.	Parameter	Description
1	Source IP Address	Client initiating traffic
2	Destination IP Address	External server accessed
3	Protocol Type	TCP, UDP, HTTPS etc.
4	Application Information	Detected application by firewall
5	VPN Status	Enabled or Disabled
6	Firewall Action	Allow or Block
7	Traffic Visibility	Visible or Encrypted

The collected data from firewall logs is saved to identify the limitations of firewall visibility after Client VPN activation and differences in the pattern of traffic.

3.6 Dataset Details

This research is based on practical experiment
The dataset consists of:

- Website access attempts
- Application traffic records
- Firewall event logs
- VPN and non-VPN traffic sessions
- Security policy enforcement results

Traffic samples are generated using common internet activities such as:

- Web browsing
- Social media access
- Video streaming
- File downloads

The dataset includes observations from both encrypted and unencrypted traffic environments.

3.7 Experimental Setup

Hardware Requirements

Table 3: Hardware Requirements

Sr.	Component	Specification
1	Firewall Device	Ubiquiti UniFi Dream Machine Pro
2	Client Computer	Windows 11/10 Workstation
3	Internet Connection	Broadband Connection
4	Network Switch	Gigabit Ethernet Switch

Software Requirements

Table 4: Software Requirements

Sr.	Software	Purpose
1	UniFi Network Controller	Firewall Management
2	VPN Client Software	VPN Connectivity
3	Wireshark	Traffic Analysis
4	Web Browser	Traffic Generation
5	Microsoft Excel	Data Analysis

3.8 Tools Utilized

The following tools are used during the experiment:

- Ubiquiti UniFi Dream Machine Pro (UDM Pro) for firewall policy management.
- VPN Client Software for creating encrypted tunnels.
- UniFi Controller Dashboard for monitoring logs and security events.
- Microsoft Excel for result comparison and visualization.

4. Experiment

Unlike a purely theoretical exercise, the results in this section come from live firewall traffic logs. Exported directly from a UDM Pro and a content-filtering policy named "Blocking Social Apps" is applied. Two client devices were each captured before and after activating a VPN client, using a different VPN service for each device. All figures below are measured values from the exported CSV logs, not simulated or illustrative estimates.

4.1 Testbed Description

- Firewall/edge device: UniFi Dream Machine Pro, with an active firewall policy blocking access to social media and video platforms (Facebook, Instagram, YouTube, TikTok, Snapchat).
- Client 1: laptop device ("TEST f5:62", IP 10.19.29.32), tested with a full-tunnel commercial VPN (Proton VPN, identified via the vpn-api.proton.me control-plane domain in the log).
- Client 2: mobile device ("Waqar-A35", IP 10.19.29.105), tested with a second Super VPN Pro service.
- Data source: UDM Pro flow-log CSV export, fields include timestamp, action (allowed/blocked), policy name, source/destination IP, destination domain, port, protocol, byte counts, and flow count.

4.2 Client#1 - Before vs. After VPN Activation (Proton VPN)

Table 5:Client 1 traffic characteristics before and after Proton VPN activation.

Metric	Before VPN	After VPN
Capture duration	5 min 33 sec	4 min 5 sec
Total flows	454	33

Blocked flows	168	0
Blocked → YouTube/Facebook hits	138	0
Unique destination IPs	170	18
Unique destination domains	148	15
DNS queries	22	9
Total bytes transferred	~10.14 MB	~84.29 MB
Bytes to single dominant IP	n/a (distributed)	84.05 MB to 198.244.141.207 (99.7% of total)

The above table clearly shows the results of client 1's traffic flows. Before VPN activation on the client device, the firewall, as per its website/apps blocking policy, blocked 168 traffic flows that contained traffic to youtube/facebook etc. And as the VPN activate in client device, the firewall blocking became 0.
It is clearly showing that all traffic goes through the VPN Tunnel and most of the traffic going on single destination (198.244.131.207), which carried 99.7% of all transferred bytes. One nuance worth noting: DNS query count and destination-IP diversity did not fall to near-zero as a simple model might predict (9 DNS queries and 18 destination IPs remained). Inspection of the remaining traffic shows it consists almost entirely of Windows OS-level connectivity checks (client.wns.windows.com, www.msftconnecttest.com, login.live.com) rather than user browsing traffic. This suggests the VPN client used split tunnelling for system-level telemetry while routing browsing traffic through the tunnel, meaning the byte-concentration signal (Layer 4) proved far more reliable here than the raw DNS-count heuristic. It is also notable that the VPN's own control-plane domain, vpn-api.proton.me, appeared directly in the log before the tunnel was fully established — meaning this session would have been caught by Layer 3 (IP/domain reputation) even before Layer 4 behavioural scoring became necessary.

4.3 Client 2 - Before vs. After VPN Activation (Second VPN Service)

Table 6:Client 2 traffic characteristics before and after activation of a second VPN service.

Metric	Before VPN	After VPN
Capture duration	4 min 56 sec	5 min 59 sec
Total flows	534	110
Blocked flows	310	0
Blocked → Facebook/Insta hits	162	0
Blocked → YouTube hits	100	0
Blocked → TikTok hits	4	0
Blocked → Snapchat hits	10	0
Unique destination IPs	79	49
DNS queries	143	16
Total bytes transferred	~7.55 MB	~5.85 MB
Bytes to single dominant IP	Distributed (top IP only 18.3% of total)	1.35 MB to top IP (~23.1% of total; no dominant tunnel)

Client 2 produced more interesting results than client1. Before VPN activation, in the client device firewall, 310 traffic flows were blocked. And the most of them were from Facebook/Instagram (162 records), YouTube (100 records) and some others. Because there is a policy of social media blocking and firewall blocks all traffic related to the policy. As the VPN is activated on the Client 2 device there is no blocking by firewall because VPN successfully bypassed the firewall policy.

The way Client 2 bypasses the firewall is different from client1. There is a different pattern of destination IP. In Client 1 where most of the traffic was concentrated on a single destination but in Client 2 scenario the traffic was spread across 49 different IP addresses and the busiest IP address carried only 23% of total traffic. Even though the traffic is routed on different destinations the VPN still completely bypass firewall.

This difference between the two VPN traffic flows shows an important pattern that all VPN don't have the same pattern of destination IP. Some of the VPN use a single destination to send most of the data by bypassing the firewall, but not all VPN.

In case 2, a different destinations are used by the VPN server.

Therefore VPN detection does not rely on single layer. We will have to capture different type of traffic.

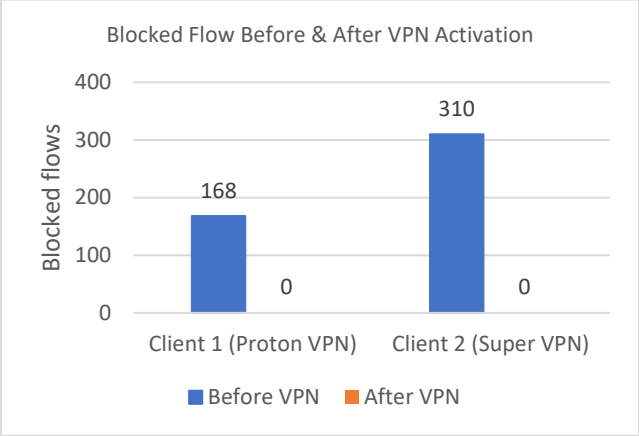

Figure 4:Blocked flows before and after VPN activation, both clients.

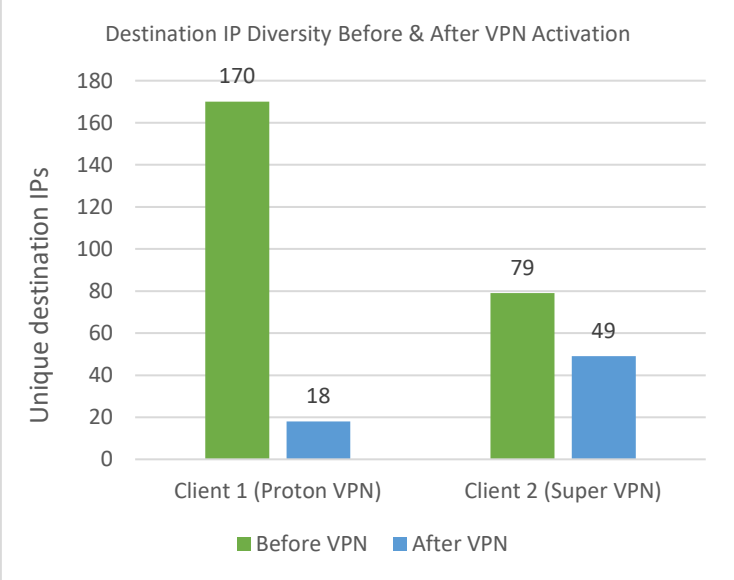

Figure 5:Unique destination IP count before and after VPN activation, both clients.

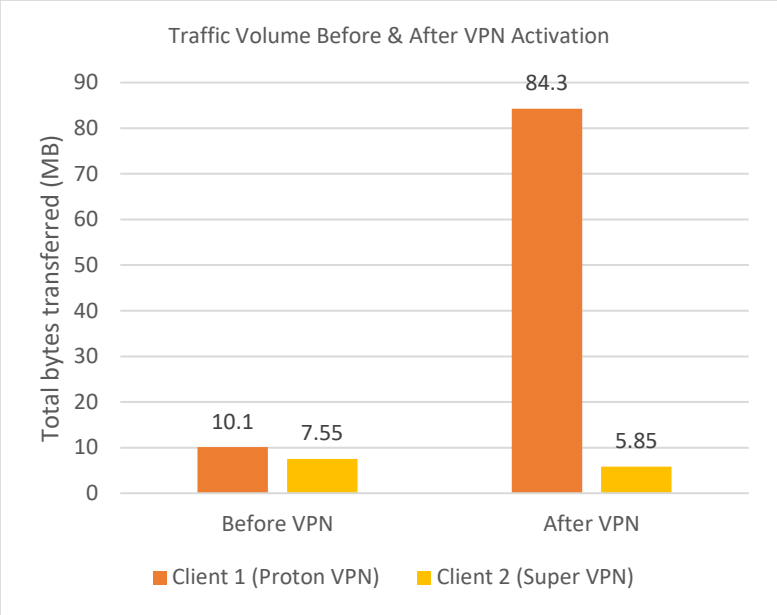

Figure 6:Total traffic volume (MB) before and after VPN activation, both clients.

4.4 Comparative Discussion

Placed side by side, these two real-world tests demonstrate that VPN effectiveness against a firewall's content-filtering policy is not uniform; it depends heavily on the specific VPN client's tunnelling architecture, and a detection framework should not assume a single universal traffic signature.

Table 7:Comparative behavioural signatures across the two tested VPN services.

Signal	Client 1 (Proton VPN)	Client 2 (Super VPN)	
Blocked-flow evasion	Complete (168 → 0)	Partial (310 → 0)	
Destination-IP collapse	Yes — 99.7% of bytes to one IP	No — traffic remained spread across 49 IPs	
DNS query drop	Moderate (22 → 9)	Moderate (143 → 16)	
VPN control-plane leak	Yes — vpn-api.proton.me visible in log	Not observed in captured domains	
Likely tunnelling model	Full-device tunnel	Full-device tunnel	

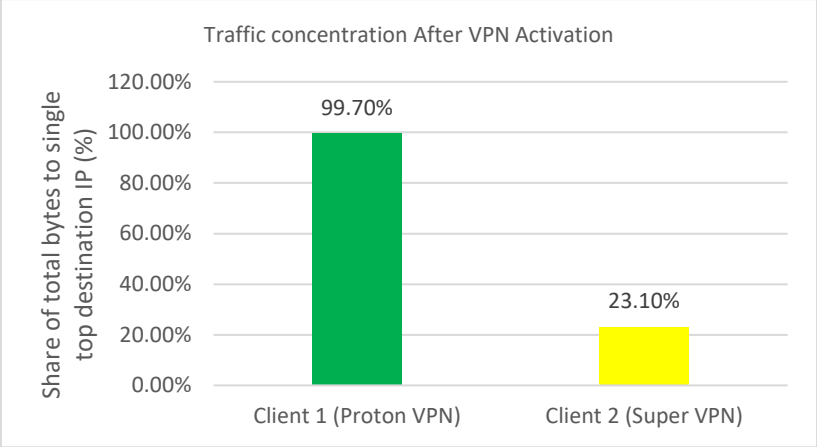

Figure 7:Share of total bytes to single top destination IP

Put side by side, Client 1 and Client 2 make an important point: getting past the firewall's content filter and leaving a "single tunnel" fingerprint in the traffic are two separate things, not two sides of the same coin. Proton VPN did both — it beat the filter completely, and it left the classic signature of one dominant destination IP. Super VPN also beat the filter completely, but it left almost no trace of that signature at all. That matters for anyone building detection around a single rule. If you only watch for traffic funnelling into one IP, you'll catch Proton VPN easily and miss Super VPN completely — even though both VPNs are equally effective at getting around the policy. This is really the core argument for treating blocked-flow behaviour and destination concentration as two separate signals in the framework, rather than assuming one implies the other. A VPN can be a total blind spot for content filtering without ever showing the traffic pattern most people expect a VPN to show.

4.5. Practical Implementation Considerations

The UDM Pro's native IDS/IPS engine already ships with several rule categories relevant to VPN protocol detection, and these can be enabled with minimal configuration. [10] TLS fingerprinting and behavioural scoring, however, generally require exporting flow logs (via syslog or NetFlow) to an external tool, even a lightweight setup such as an ELK stack or a custom script, since the UDM Pro's own interface does not natively perform this kind of correlation, as demonstrated by the manual CSV analysis used to produce Section 4 of this paper. DNS-layer visibility can be improved further by routing DNS through a filtering or logging service, which also helps catch the installation phase of VPN apps.

A practical takeaway from Section 4.4 is that organizations should not assume all VPN evasion looks the same. A single detection rule tuned against one VPN client's behaviour (such as Client 1's full-tunnel collapse) would give a false sense of coverage, since a different VPN client (Client 2) evaded detection under that same rule despite being just as fully effective at bypassing the content policy as Client 1.

4.6. Proposed Detection Framework

We propose a layered framework, since no single signal is reliable on its own, legitimate business VPNs to partners or remote offices would otherwise be flagged as false positives, and as Section 4.4 shows, not all unauthorized VPN clients behave the same way either.

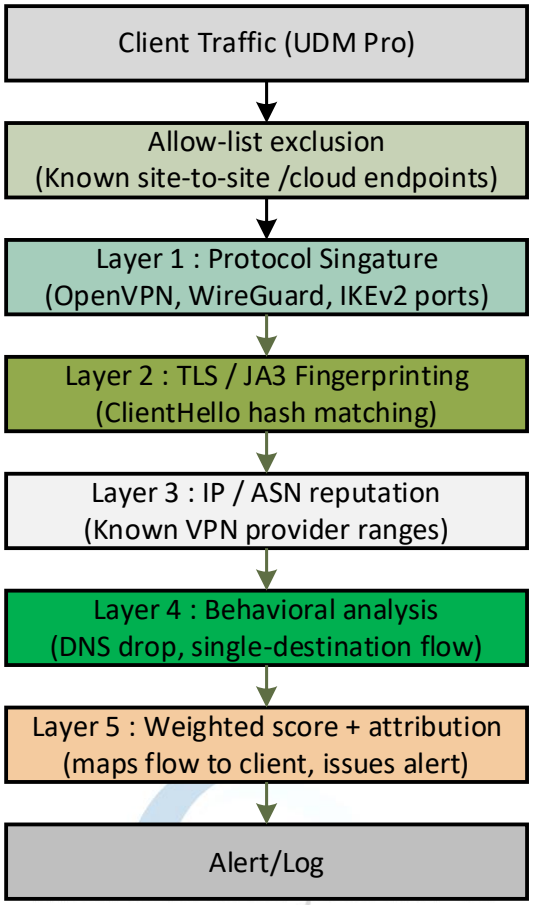

Figure 8: 5-Layer Framework

Layer 1 — Protocol signature matching

Flag traffic that is matching known VPN protocol ports and signatures
1. OpenVPN on UDP/TCP 1194
2. Wire Guard on UDP 51820
3. IKEv2 on UDP 500/4500
and so on.
This catches unsophisticated VPN use immediately and can be enabled through UDM Pro.

Layer 2 — TLS fingerprinting

Need to detect VPN Traffic that is inside HTTPS traffic, for traffic tunnelled over port 443 specifically to evade Layer 1. Many VPN use TCP port 443 because most of network allow HTTPS. Compute a JA3 hash from each new TLS handshake and compare it against a maintained list of fingerprints known to belong to common VPN clients [11][16][17].

Layer 3 — IP and ASN reputation

In this layer identify that the traffic is belongs to a known VPN provider's destination. Cross-reference destination IP addresses against known VPN provider address ranges and autonomous system numbers. [9]. In our own captured data (Section 4.2), this layer alone would have caught the Proton VPN session, since its control-plane domain (vpn-api.proton.me) appeared directly in the firewall log before the tunnel masked further traffic.

Layer 4 — Behavioural analysis

For traffic that evades the first three layers, look for the behavioural pattern described:
1. Flow duration
2. Packets sizes
3. Number of Packets
4. Bytes transferred
5. Session Length/long-lived
6. Destination diversity
7. Byte Concentration
Section 4.4 shows this signal held strongly for one VPN client in our test and did not hold at

all for the other, which is an important caveat for relying on this layer alone [13].

Layer 5 — Client attribution

Once a flow is flagged by any of the above, correlate the source IP back to a specific device using the UDM Pro's own client list and DHCP lease information, producing a human-readable alert naming the device and, where available, the associated user [14][15].

4.7.Limitations

This study does not yet have an example of a VPN client that evades content-policy blocking only partially, a case worth testing in future work to validate that the framework's layered scoring also performs well against intermediate evasion behaviour.

In session 4.6 the proposed framework is have some limitations. It cannot decrypt or inspect tunnel content, so it can only establish likelihood, not certainty, of bypass activity. Confidence thresholds require local tuning against an organization's actual traffic mix to avoid false positives on legitimate encrypted business traffic. Commercial VPN IP ranges and JA3 fingerprints require ongoing maintenance, since providers update client software and rotate infrastructure [9], and browser-side TLS extension randomization is already eroding JA3's reliability as a standalone signal [8]. The experimental captures in Section 4.1-4.3 covered only two client devices and two VPN services over short capture windows (under seven minutes each); a production deployment should validate the framework against a broader and longer-running sample before relying on it for enforcement decisions.

4.8. Excluding Legitimate Corporate VPN Traffic

Such a detection mechanism is useful only if it can distinguish between an employee attempting to bypass content rules and the organization's own legitimate VPN traffic (such as site-to-site tunnels between branch offices, remote access to servers by administrators, or applications accessing cloud services-AWS, Azure, GCP-via a VPN or private connection). The challenge is that this traffic closely resembles the bypass traffic described earlier: it is long-lived, directed at a single external IP address, and carries encrypted, high-entropy data. Without accounting for this, the system would constantly flag traffic that is critical to the organization's operations.

To address this, we treat known corporate VPN endpoints as an "allow-list" and check this list before any of the five detection layers are triggered:

1. Exclusion of known endpoints like Site-to-site tunnels and cloud connections typically terminate at a small, fixed set of IPs or ASNs. Unlike commercial VPN exit nodes, these do not change, allowing them to be maintained in a static allow-list.
2. Device and resource identification: legitimate corporate VPN tunnels usually originate from known infrastructure such as a dedicated gateway or a UDM Pro's site-to-site VPN setup rather than from a random employee's laptop.
3. Protocol and configuration awareness: Since the organization configures its own site-to-site VPN, it already knows which protocols and configurations to expect; consequently, there is no need to rely on guesswork via fingerprinting or scoring methods.
4. Directionality as a signal: While enterprise VPN and cloud connections are typically initiated by servers or infrastructure devices, the use of a VPN for circumvention usually begins with a regular employee accessing a consumer VPN provider from their workstation.

This five-layer system inspects only standard end-user device traffic destined for locations outside the organization's approved VPN infrastructure. This enables the system to detect unauthorized bypass attempts without disrupting processes essential to the organization, such as server access, cloud connections, or inter-office links.

5. Conclusion

The issue of bypassing firewalls via Third-party/Open-Source VPNs can't be resolved by using only Blocklist, port rules. The result of both clients with different VPNs shows that every VPN uses a different traffic pattern for VPN tunnels.

This research shows that by combining the proposed layer-based framework.

1. VPN Protocol Signatures

2. TLS Fingerprinting
3. IP reputation analysis
4. Network flow behavioural analysis
5. Client Contribution
6. Allow List

An organization using the UDM Pro can improve firewall visibility by detecting traffic patterns of VPN traffic and, with an allow-list improve false-flag alerts from company site-to-site VPNs and remote server access to ensure normal operations.

REFERENCES

M. Conti, A. Dehghantanha, K. Franke, and S. Watson, "Internet of Things security and forensics: Challenges and opportunities," Future Generation Computer Systems, vol. 78, pp. 544–546, 2021.

National Institute of Standards and Technology (NIST), "Zero Trust Architecture," NIST SP 800-207, 2020.

Cisco, "Annual Internet Report (2018–2023) White Paper, 2020".

I. Alwhbi, C. Zou, and R. Alharbi, "Encrypted Network Traffic Analysis and Classification Utilizing Machine Learning," Sensors, vol. 24, 2024, Art. no. 3509, doi: 10.3390/s24113509.

D. Berman et al., "A Survey of Deep Learning Methods for Cyber Security," Information, 2021.

M. Lopez-Martin et al., "Deep Learning for Network Traffic Classification," IEEE Access, 2023.

Y. Zhang et al., "Network Traffic Analysis Using Machine Learning Techniques in Encrypted Environments," Journal of Network and Computer Applications, 2024.

Stamus Networks, "Adapting to change: JA3 fingerprints fade as browsers embrace TLS extension randomization," Stamus Networks Blog, Jan. 10, 2024. [Online]. Available: https://www.stamus-networks.com/blog/ja3-fingerprints-fade-browsers-embrace-tls-extension-randomization

M. T. Khan, J. DeBlasio, G. M. Voelker, A. C. Snoeren, C. Kanich, and N. Vallina-Rodriguez, "An empirical analysis of the commercial VPN ecosystem," in Proc. 2018 Internet Measurement Conference (IMC '18), Boston, MA, USA, 2018, pp. 443–456. doi: 10.1145/3278532.3278570.

Ubiquiti Inc., "UniFi Dream Machine ProThreat Management and IDS/IPS," Ubiquiti Help Center. Accessed: Jul. 10, 2026.

J. Althouse, J. Atkinson, J. Atkins, "JA3 - a method for profiling SSL/TLS clients," Salesforce Engineering Blog, 2017.

Draper-Gil, G., Lashkari, A. H., Mamun, M. S. I., & Ghorbani, A. A. "Characterization of Encrypted and VPN Traffic using Time-related Features," *Proc. 2nd Int. Conf. on Information Systems Security and Privacy (ICISSP 2016)*, Rome, Italy, pp. 407–414, 2016.

Hanlon, M. et al. "Detecting VPN Traffic through Encapsulated TCP Behavior," *Privacy Enhancing Technologies Symposium (FOCI 2024)*.

Draper-Gil et al. work cited via "VPN Traffic Analysis: A Survey on Detection and Application Identification," *IEEE Access*, 2026.

Afuwape et al., cited in "Network traffic classification: Techniques, datasets, and challenges," *Computer Networks*, ScienceDirect, 2022.

Matoušek, P. et al. "On Reliability of JA3 Hashes for Fingerprinting Mobile Applications," Springer/EUDL, 2021.

"A Survey on TLS-Encrypted Malware Network Traffic Analysis Applicable to Security Operations Centers," *Applied Sciences (MDPI)*, 2021.

Začs, A. "Analysis of VPN Obfuscation Methods," conference paper. "Next-Generation Firewalls: Beyond Traditional Perimeter Security," International Journal for Multidisciplinary Research (IJFMR), 2025.

K. Hamid, M. W. Iqbal, M. Aqeel, X. Liu, and M. Arif, "Analysis of Techniques for Detection and Removal of Zero-Day Attacks (ZDA)," in Proc. [conference/proceedings — verify exact venue/year on ResearchGate], The Superior University, Lahore, Pakistan.

K. Hamid, F. Naeem, M. Ibrar, A. M. Delshadi, F. Ahmed, M. Aamir, and G. Ahmad, "Behavior and Characteristics of Ransomware – A Survey," The Superior University, Lahore, Pakistan.

K. Hamid, M. W. Iqbal, M. Aqeel, T. A. Rana, and M. Arif, "Cyber Security: Analysis for Detection and Removal of Zero-Day Attacks (ZDA)," in Artificial Intelligence & Blockchain in Cyber Physical Systems, CRC Press, 2023.

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.

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.

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

zishan zafar, K. Hamid, M. Kafayat, M. waseem Iqbal, Z. Nazir, and A. Ghani, "AI-Based Cryptographical Framework Empowered Network Security," *Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)*, vol. 42, pp. 497–510, Apr. 2023, doi: 10.17605/OSF.IO/W69VT.

K. Hamid *et al.*, "Empowering Robust Security Measures in Node.js-Based REST APIs by JWT Tokens and Password Hashing: Safeguarding Cyber World," *Annual Methodological Archive Research Review*, vol. 3, May 2025, doi: 10.63075/w2nam443.

M. Delshadi *et al.*, "Empowerment of Artificial Intelligence (AI) In Preventing and Detecting Ransomware: An Analytical Review," *Spectrum of Engineering Sciences*, vol. 3, no. 9, pp. 36–48, Sep. 2025

M. Delshadi, M. Zubair, K. Hamid*, and M. W. Iqbal, "Preventing and Detecting Malicious Activities During Phishing Attacks Using AI-Based Sandbox ByPass," *Annual Methodological Archive Research Review*, vol. 3, no. 8, pp. 249–261, Aug. 2025, doi: 10.63075/fahtkz35.

Ahmad, M. N. Amin, K. Hamid, S. M. Rizwan, and S. A. A. Naqvi, "Enhanced IoT Network Security for Network intrusion detection Model based on Machine Learning Technique," *Annual Methodological Archive Research Review*, vol. 3, no. 8, Aug. 2025, doi: 10.63075/0vcg0093.

M. H. Akhtar, T. Jabeen, R. Aziz, M. N. Amin, S. M. Rizwan, and K. Hamid, "Intelligence based Self-Healing Network Design: An Automated Incident Response System for Troubleshooting of IoT Security Breaches," *Annual Methodological Archive Research Review*, vol. 3, no. 8, Aug. 2025, doi: 10.63075/6dhhj119

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.

Aslam, W. Tariq, T. Waheed, K. Hamid, S. M. Rizwan, and A. Ahmed, "Enhancing Privacy and Security in Multi-Party Computation for Data Mining in Cryptographically Protected Environments," *Annual Methodological Archive Research Review*, vol. 3, no. 8, pp. 510–531, Aug. 2025, doi: 10.63075/fxe5gg65.

M. D. Rasheed, K. Hamid, M. W. Iqbal, M. W. Iqbal, M. Ibrar, and M. A. Khan, "Secure Evolutionary Model Empowered Smart Homes in AI Environment : An Efficient Way of Life," 한국차세대컴퓨팅학회 학술대회, pp. 254–257, Dec. 2025.