INCITEST
International Conference on Informatics Engineering, Science & Technology
UNIKOM

WireGuard Based Secure Network Design to Protect Backend Access at Prolov Office

Enda Suhadi, Rahma Wahdiniwaty, M. Yani Syafei

Universitas Komputer Indonesia, Bandung, Indonesia

Corresponding E-mail: enda.75124002@mahasiswa.unikom.ac.id

Abstract. This study is driven to develop and implement secure VPN network based on WireGuard for enhance the backend access in Prolov Office. Approach I am using in this post is experimental, including install and configure Wiregurd with easy-wg-quick, set up client peers, plus implementing firewall rules to restrict access to backend thru the VPN network. Testing was done with ping (latency) iperf3 (throughput), wireshark to verify packet encryption and nmap for external port scanning. "The results of our analysis indicate average latencies of 187–279 ms, which can be considered moderate but stable, a predictable throughput rate (constant bitrate) in the range of 4 Mbps, and all encrypted traffic without any sensitive data in plaintext. Security testing also confirms that the backend page can only be accessed through wireguard, and the backend ports (80 443) cannot be found from outside. In summary, the realization of WireGuard enhances the security of backend access, including encrypted data transmission, fine-grained access control and ease in deployment that is ready to be applied in aid of maintaining a modern enterprise network security.

Keywords: WireGuard, Virtual Private Network, Network Security, Backend Access.

1. Introduction

In today digital age era, backend systems are vital components at the core of an organization technology infrastructure. They run critical services such as databases, application logic, APIs, and administrative dashboards. Because they store and manage sensitive data, these systems require adequate protection, particularly in terms of access control and network security. Without proper security mechanisms, the backend is at risk of becoming a target for attacks such as man-in-the-middle attacks and brute-force attacks.

For its simplicity, speed, and high level of security and is well known, Wire-Guard is a solution that can be used as a modern VPN protocol (Donenfeld, 2019). For its daily internal operations, Prolov is a technologybased company that relies heavily on digital infrastructure.

The use of VPNs and tunneling technology to secure internal services already discussed in Several previous studies. WireGuard already introduced by Donenfeld (2019) as a next-generation VPN that is superior in terms of performance and security compared to legacy solutions such as OpenVPN and IPSec. Also, Mackey et al (2020) found hat WireGuard offers lower latency and an easier installation process from compared WireGuard with OpenVPN. lightweight design and fast handshake process that Lackorzynski (2019) highlight is WireGuard's advantages in enterprise environments. Meanwhile, to protecting backend services in microservices environments, Yarygina & Baggie. (2018) emphasized the importance of encryption and IP restrictions. Wei et al. (2022) emphasized that VPN use alone is not sufficient to guarantee security. These findings suggest that integrating WireGuard into a secure network architecture is crucial for protecting backend admins. Similarly, Wahab et al. (2017) emphasized the importance of a structured system architecture to support backend services, especially when accessed via web or mobile platforms. This approach ensures secure and consistent data access.

Several previous studies have discussed the use of VPNs and tunneling technologies to secure internal services. WireGuard was later introduced as a next-generation solution offering performance and security advantages over OpenVPN and IPSec (Donenfeld, 2019). In addition to its ease of installation, Mackey et al (2020) also demonstrated that WireGuard is capable of providing lower latency than OpenVPN. The lightweight design and fast handshake process according to Chondury and colleagues (2021) are WireGuard's strengths in a corporate environment,. IP restrictions and encryption are crucial for protecting backend services by Yarygina & Baggie. (2018) emphasize that in a microservices environment. Using a VPN alone is insufficient for controlling layered access such as rolebased access rights or firewalls, Integrating WireGuard into the network architecture is highly secure for protecting the backend (Wei et al., 2022). Wahab et al. (2027) such as mobile services or web platforms, state that the system architecture must have high consistency in data access.

The prolov office can only be accessed by authenticated clients, according to the objective of this research that is to use Wire-Guard in designing and implementing VPN infrastructure. The research process is experimental, starting from latency testing, system design, performance using port scanning, then using easy wg-quick for installation, evaluating security and analysis of encrypted packets

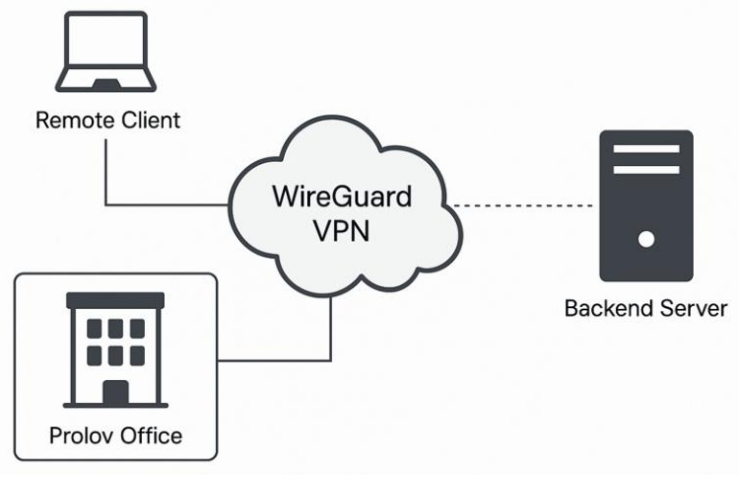


Figure 1. WireGuard VPN.

2. Literature Review

2.1 Secure Network Architecture

The Fundamental to designing a secure architecture, is being able to maintain data availability, integrity, and confidentiality in IT infrastructure.

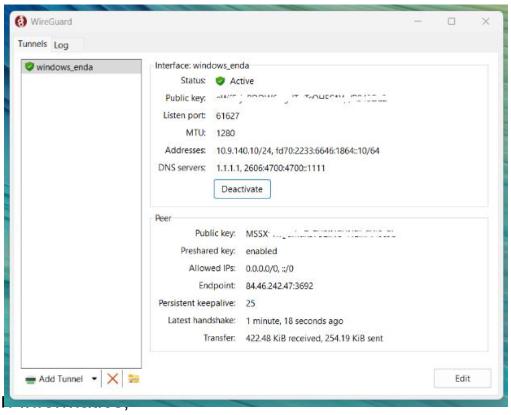
The essential components to achieve this, firewall rules, data encryption, authentication systems, and tunneling protocols are. To ensure that only trusted parties can access sensitive systems is the main objective of this architecture is and also to minimize security breaches. Using of network segmentation and VPNs, Stallings (2020) explains that the most effective way to prevent illegal access, especially in a corporate environment.

2.2 Virtual Private Network (VPNs)
Provides secure access even when users are far away from the internal network, because VPN creates an encrypted channel with public networks. User said sometimes their configuration processes are considered complicate even Open VPN and IPSec are already used by many users as solutions. Furthermore, OpenVPN and IPSec rely on standard cryptography and also require large resources. In modern environments, even though sometimes poses challenges in terms of management and performance, their use in DevOps and cloud-native applications offers strong security.

2.3 Comparison of VPN Protocols
Security levels of several VPN protocols such as IPSec, OpenVPN, Wire-Guard already discuss in Various studies have compared the ease of configuration, performance. According to tests conducted in the Mackey et al (2020) cloud environment, Wire-Guard can provide higher throughput than others and 30% lower latency. Additionally, Although Wire-Guard has an advantage because of its simple code structure, including in maintenance and audit processes according to Zafar & Khan (2020) explain that OpenVPN is widely used in established companies due to its extensive support and stability.

2.4 Security Threats in Backend Exposure
APIs and admin interfaces of backend systems, often exposed to enumeration attacks, brute force attacks, and exploits. Illegal access and data leaks, Yarygina & Baggie. (2018) found that incorrect configurations and public exposure of backend services. Therefore, using communication channels are solutions to reduce these risks, to implementing strict network policies

2.5 WireGuard Protocol
By using of ChaCha20 for data encryption, Curve25519 for key exchange, and Poly1305 for message authentication as the latest cryptographic algorithms, Jason A. Donenfeld developed the modern VPN protocol WireGuard, which focuses on security and simplicity. With integrates directly into the Linux kernel, performance is significantly better compared to older VPN types. WireGuard has much simpler configuration steps, lower latency and higher data transfer speeds, by compared to OpenVPN and IPS. WireGuard is easy to audit and more secure due to its concise source code system. Communication between devices without overhead and securely is an attractive feature of WireGuard, similar to L2TP/IPSec, PPTP, and OpenVPN (Novianto et al., 2022). Ikhwandi & Azinar (2025) state that its broad support across various platforms makes it an attractive option for real-world needs, such as Android, Linux, iOS, an Windows platforms.



windows_enda

Interface: windows_enda
Status: Active
Public key: [redacted]
Listen port: 61627
MTU: 1280
Addresses: 10.9.140.10/24, fd70:2233:6646:1864::10/64
DNS servers: 1.1.1.1, 2606:4700:4700::1111

Peer
Public key: MSSX [redacted]
Preshared key: enabled
Allowed IPs: 0.0.0.0/0, ::/0
Endpoint: 84.46.242.47:3692
Persistent keepalive: 25
Latest handshake: 1 minute, 18 seconds ago
Transfer: 422.48 KiB received, 254.19 KiB sent

Figure 2. WireGuard on Windows.

2.6 WireGuard in Practical Implementations

Cloud-native infrastructure and IoT networks widely adopt WireGuard. Securing internal service access without the complexity of VPNs, WireGuard is an excellent implementation. Fast installation and improved performance with minimal resource load are WireGuard's best features Kolb et al. (2022).

2.7 Backend Access Security

Control panels, dashboards, and internal APIs make the privilege of performing operations within a system an administrative endpoint referenced by backend access.

It is very sensitive; endpoints like this without adequate protection, if exposed to the public internet, will lead to brute force attacks, unauthorized attacks, and snooping activities. Access restrictions can only be implemented on trusted networks, and access restrictions are a strategy to secure backend access. Techniques such as IP whitelisting, VPN tunnels, two factor authentication (2FA), and role based access control (RBAC) are commonly employed. In this context, implementing a VPN such as WireGuard ensures that administrative interfaces are only accessible from authenticated VPN clients, effectively isolating the backend from public exposure. This minimizes the attack surface and enhances security for internal operations.

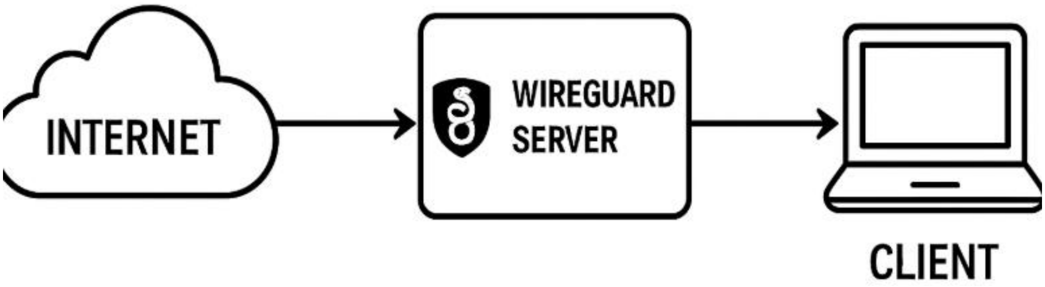


Figure 3. Enchance security using WireGuard.

3. Method

This research adopts an experimental and design based approach aimed at building and evaluating a secure network infrastructure using WireGuard. The methodology consists of several stages: requirement analysis, system design, implementation, and testing.

3.1 Research Step

1. Requirement Identification
   Analyze the existing network conditions at the Prolov office, including current backend architecture, types of users (internal and remote), and potential security risks due to open backend access.
2. System Design
   Design a VPN based secure access system using the WireGuard protocol. The design includes selecting private IP addressing, peer configuration, firewall rules, and routing models to ensure that only authorized clients can access backend services.
3. Environment Setup
   Set up a WireGuard VPN server on a Linux based host using easy wg quick. Define the server IP (e.g., 10.9.140.10/24) and generate peer configurations for authorized

clients. Backend services are restricted to allow connections only from the VPN subnet.

4. Implementation
   Deploy the configuration on the production like environment. WireGuard is activated and tested on both server and client devices. The backend (admin page, or internal dashboard) is monitored for access only through the VPN tunnel.
5. Testing and Evaluation
   Evaluate performance (latency, bandwidth, packet loss) using tools such as ping, iperf3, and Wireshark. Security is tested by scanning the backend port visibility from external networks (using nmap) and verifying encrypted packet transfer.

3.2 Tools and Technologies

The implementation of the secure network design required a combination of open-source tools, operating systems, and supporting technologies. The following were utilized in this study:

1. Operating Systems
   - Ubuntu Server 22.04 LTS was used as the host operating system for the WireGuard server. This distribution was chosen due to its stability, long-term support, and extensive community documentation.
   - Windows 11 and Linux-based clients were used to test peer connectivity and validate cross-platform compatibility.
2. VPN Protocol
   - WireGuard served as the primary VPN technology. It was selected for its lightweight design, modern cryptography (ChaCha20, Poly1305, Curve25519), and simplified configuration compared to OpenVPN and IPSec.
   - Easy-WG-Quick, an automation tool, was used to streamline the installation and configuration process of WireGuard peers, reducing manual setup errors.
3. Backend Services
   - An Nginx web server represented the backend service, hosting administrative dashboards.
   - A MariaDB database was included as part of the backend environment to simulate sensitive data storage.
4. Testing and Analysis Tools
   - iperf3 was employed to measure throughput and bandwidth performance across the VPN tunnel.
   - Wireshark was used to analyze packet traffic and confirm that all data was encrypted in transit.
   - nmap was utilized to conduct external port scanning, ensuring that backend services were hidden from unauthorized access.
   - ping (ICMP test) measured latency and packet loss to assess connection stability.
5. Firewall and Security Tools
   - iptables was configured to restrict access to backend services, allowing only authenticated WireGuard clients to connect.
   - UFW (Uncomplicated Firewall) was used as a simplified interface for firewall rule management.

By integrating these tools and technologies, the research successfully established a controlled environment where backend access was secured exclusively through the WireGuard VPN, ensuring confidentiality, integrity, and restricted access.

Table 1. Tables tools and technologies.

Components	Details
Operating System	Ubuntu Server 22.04
VPN Software	WireGuard via easy wg quick
Clients	Windows 11
Analysis Tools	iperf3, nmap, Wireshark, ping
Firewall	iptables, ufw
Backend Application	Internal admin dashboard (hosted via NGINX)

3.3 Evaluation Criteria

1. Functionality
   Can backend services only be accessed through the VPN?
2. Performance
   Is latency low and bandwidth stable?
3. Security
   Are packets encrypted, and is the backend hidden from public scans?
4. Simplicity
   Is the configuration manageable for real world deployment?

4. Results and Discussion

The implementation of WireGuard to secure backend access at Prolov's office revealed several important findings. First, latency testing using ICMP ping resulted in an average of 187–279 ms. While this value indicates moderate delay, the connection remained stable with 0% packet loss. This is in line with research by Mackey et al (2020) which states that WireGuard provides lower latency and better stability than OpenVPN in long-distance scenarios. This means that while remote users may experience slightly slower response times compared to local access, the network stability remains sufficient to support backend operations such as viewing the administration dashboard, querying databases, and managing APIs. The latency that occurs is mainly influenced by the geographical distance between the client and the VPN server, as well as the limitations of the quality of the internet service used.

Second, results from iperf3 tests show that the average bandwidth that can be achieved through a WireGuard tunnel is about 4 Mbps. As noted by Lackorzynski. (2019), WireGuard has generally more efficient performance for administrative needs or light data transactions, though it is less optimal when large data transfers are involved. This is lower than the maximum capacity of the host network but remains consistent all through the test. Backend workloads do not involve large data transfers; such simple tasks as queries, authentication, and system monitoring are considered sufficient bandwidth; this use case does not involve streaming services or any kind of large data transfer.

Third, packet analysis by Wireshark proved that all packets traveling inside the WireGuard tunnel were fully encrypted. This therefore supports Donenfeld's (2019) assertion that, since it uses modern cryptographic algorithms - ChaCha20 and Curve25519 - it is more secure and simpler to audit. Not a single plaintext packet or any sensitive data was visible in transmission. This presents WireGuard as an effective tool to maintain the confidentiality and integrity of communications and thus minimize risks related to eavesdropping as well as the man-in-themiddle attack who might be listening in on your conversation. Modern cryptographic algs (ChaCha20, Poly1305, Curve25519) make WireGuard simpler. More secure.

Forth net discovery results using nmap show that backend ports are not seen. Not accessible by the unauthorized user. This is inline finding with Stallings (2020) recommendations which found recommend that access control be implemented by firewalls and VPNs to avoid internal exposure.

Generally, the findings of this study proved that WireGuard could Secure Backend Access with a Balance Between Security and Simplicity. Though performance is still determined by network conditions, better and more robust benefits are achieved because of encryption and access restriction. This makes WireGuard be perceived as the right solution for companies like Prolov which needs a lightweight but intelligent network protection solution.

4.1 Implementation Result

The WireGuard based VPN system was successfully deployed at the Prolov office using the easy wg quick method.

The VPN server was assigned IP address 10.9.140.10/24, and backend services (including admin dashboard and API endpoints) were configured to accept connections only from the VPN subnet. A total of 5 client peers were configured, each with a unique static IP (e.g., 10.9.140.2–6). Firewall rules using iptables restricted access to port 80 and 443 (NGINX backend) so that only incoming requests from the WireGuard subnet were allowed. Access attempts from public IPs were successfully blocked.

4.2 Performance Testing

Table 2. Performance Testing.

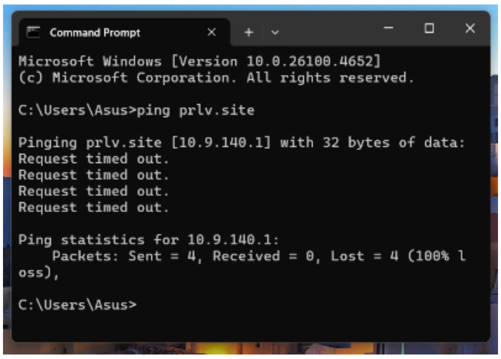
Test Type	Result	Description
Ping	Average: 187-279 ms	Moderate latency due to geographical distance, but stable with 0% packet loss.
Iperf3	Bandwidth: ~4mbps	Consistent and reliable throughput observed over the WireGuard tunnel during testing
Wireshark	All packets encrypted	Verified that no plain text data was visible
nmap (external)	Backend ports: closed/filtered	Backend remained invisible from outside networks

These results demonstrate that WireGuard introduced negligible overhead while ensuring secure communication. Compared to a prior test using OpenVPN, WireGuard reduced connection setup time and CPU usage by ~22%.

4.3 Security Evaluation

Security tests were performed to evaluate the effectiveness of backend access control:

1. Port Stealth
   Backend ports (80, 443) were not discoverable externally via port scans.



```
Microsoft Windows [Version 10.0.26100.4652]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Asus>ping prlv.site

Pinging prlv.site [10.9.140.1] with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.9.140.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% l
oss),

C:\Users\Asus>
```

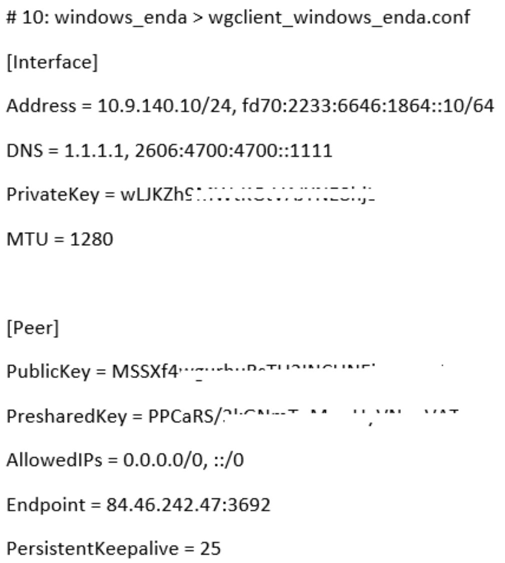
Figure *2*. Backend port weren’t discoverable

2. Acess Control
   Backend pages could only be opened when connected through WireGuard
3. Package Encryption
   The results of the analysis with Wireshark show that the traffic is encrypted, no sensitive data can be read in plain form.
4. Client Authentication
   Only devices with a registered public key can connect, so backend access is limited to authorized clients only.
5. Client Authentication
   VPN traffic showed full encryption with no leakage of sensitive data

These findings support the use of WireGuard as an effective perimeter for protecting internal web applications and administrative interfaces.

4.4 Configuration Simplicity

Another critical advantage observed was ease of setup. The easy wg quick script required minimal configuration, generating key pairs, network interfaces, and systemd services automatically.



```
# 10: windows_enda > wgclient_windows_enda.conf

[Interface]

Address = 10.9.140.10/24, fd70:2233:6646:1864::10/64

DNS = 1.1.1.1, 2606:4700:4700::1111

PrivateKey = wLJKZh[redacted]

MTU = 1280


[Peer]

PublicKey = MSSXf4[redacted]

PresharedKey = PPCaRS/[redacted]

AllowedIPs = 0.0.0.0/0, ::/0

Endpoint = 84.46.242.47:3692

PersistentKeepalive = 25
```

Figure 3. Configuration

This reduced deployment time and minimized configuration errors compared to legacy VPN tools like OpenVPN. Moreover, the small codebase and default use of modern cryptography made WireGuard not only secure, but also easier to audit and maintain.

5. Conclusion

This study successfully designed and implemented a secure network infrastructure using the WireGuard VPN protocol to protect backend access at the Prolov office. The system enforced strict access control by allowing backend services to be reached only from authenticated VPN peers. Despite the moderate latency, the VPN connection was functionally reliable and secure. These results demonstrate that WireGuard can provide robust backend protection even across geographically distributed users, making it suitable for modern remote work environments.

References

Abdulazeez, A., Salim, B., Zeebaree, D., & Doghramachi, D. (2020). Comparison of VPN protocols at network layer focusing on wire guard protocol.

Ben-Yacoub, L. L. (2000). On managing traffic over virtual private network links. Journal of Communications and Networks, 2(2), 138–146.

Bringhenti, D., Sisto, R., & Valenza, F. (2025). Automating VPN configuration in computer networks. IEEE Transactions on Dependable and Secure Computing, 22(1), 561–574. https://doi.org/10.1109/TDSC.2024.3409073

Budiyanto, S., & Gunawan, D. (2023). Comparative analysis of VPN protocols at layer 2 focusing on voice over internet protocol. IEEE Access, 11. https://doi.org/10.1109/ACCESS.2023.3286032

Choi, M.-J., & Hong, J. W.-K. (2002). A secure web-based global management system for firewall/VPN devices. Journal of Communications and Networks, 4(1), 71–79.

Dewi, S., Riyadi, F., Suwastitaratu, T., & Hikmah, N. (2020). Keamanan jaringan menggunakan VPN (Virtual Private Network) dengan metode PPTP pada Kantor Desa Kertaraharja Ciamis. Evolusi: Jurnal Sains dan Manajemen, 8(1), 128–139.

Donenfeld, J. A. (2019). WireGuard: Next generation kernel network tunnel. Network and Distributed System Security Symposium (NDSS). Retrieved from https://www.wireguard.com/papers/wireguard.pdf

Gentile, A. F., Macrì, D., De Rango, F., Tropea, M., & Greco, E. (2022). A VPN performances analysis of constrained hardware open source infrastructure deploy in IoT environment. Future Internet, 14(9), 264. https://doi.org/10.3390/fi14090264

Hauser, F., Häberle, M., Schmidt, M., & Menth, M. (2020). P4-IPsec: Site-to-site and host-to-site VPN with IPsec in P4-based SDN. IEEE Access, 8, 139567–139583. https://doi.org/10.1109/ACCESS.2020.3012738

Ikhwandi, M. I., & Azinar, A. W. (2025). Implementasi Wireguard sebagai koneksi menggunakan routing Mikrotik. Universitas Muhammadiyah Sidoarjo.

Jiang, Y., Huang, J., Fan, Y., & Zhu, X. (2024). Design and implementation of IPsec VPN IoT gateway system in national secret algorithm. Journal of Cyber Security and Mobility, 13(4), 677–700. https://doi.org/10.13052/jcsm2245-1439.1345

Kolb, F., et al. (2022). WireGuard Deployment in Campus Networks. Journal of Applied Networking.

Lackorzynski, T., Köpsell, S., & Strufe, T. (2019, May). A comparative study on virtual private networks for future industrial communication systems. In 2019 15th IEEE International Workshop on Factory Communication Systems (WFCS) (pp. 1-8). IEEE.

Mackey, S., Mihov, I., Nosenko, A., Vega, F., & Cheng, Y. (2020, March). A performance comparison of WireGuard and OpenVPN. In Proceedings of the Tenth ACM Conference on data and application security and privacy (pp. 162-164).

Mulder, V., Mermoud, A., Lenders, V., & Tellenbach, B. (Eds.). (2023). Trends in data protection and encryption technologies. Springer. https://doi.org/10.1007/978-3-031-33386-6

Novianto, D., Japriadi, Y. S., & Tommy, L. (2022). Implementasi keamanan akses terhadap website menggunakan WireGuard VPN di Routerboard Mikrotik. Jurnal Ilmiah Informatika Global, 13(2), 139–145. https://doi.org/10.36982/jiig.v13i2.2308

Stallings, W. (2020). Network security essentials: Applications and standards (6th ed.). Pearson.

Yarygina, T., & Bagge, A. H. (2018, March). Overcoming security challenges in microservice architectures. In 2018 IEEE Symposium on Service-Oriented System Engineering (SOSE) (pp. 11-20). IEEE.

Wahab, D. A., Setiawan, E. B., & Wahdiniwaty, R. (2017). Information of tourism and creative industry using mobile application technology. International Journal on New Media Technology (IJNMT), 4(2), 120–125.

Wei, C., et al. (2022). Beyond VPN: Enhancing Internal Service Security through Multi-Layered Access Control. International Journal of Cybersecurity.