

Security of VPNs in High-Surveillance Environments: A Comparative Study of VPN Alternatives

Anri HASANI

EUROPEAN UNIVERSITY OF TIRANA, FACULTY OF ENGINEERING,
INFORMATICS AND ARCHITECTURE, DEPARTMENT OF INFORMATICS
AND TECHNOLOGY, TIRANA, ALBANIA
ahasani5@uet.edu.al

Malvina NIKLEKAJ

EUROPEAN UNIVERSITY OF TIRANA, FACULTY OF ENGINEERING,
INFORMATICS AND ARCHITECTURE, DEPARTMENT OF INFORMATICS
AND TECHNOLOGY, TIRANA, ALBANIA
malvina.niklekaj@uet.edu.al

Abstract

Virtual Private Networks (VPNs) play a crucial role in ensuring secure communication over public networks. They are widely used for protecting online privacy, circumventing censorship, and enabling secure remote access to networks. However, despite their increasing adoption, VPNs face significant security vulnerabilities, misconfigurations, and performance-related challenges, particularly in high-surveillance environments. The growing sophistication of surveillance technologies, such as deep packet inspection (DPI) and metadata analysis has made it increasingly difficult for VPNs to provide true anonymity and confidentiality.

This paper provides a comprehensive analysis of VPN security, examining traditional protocols such as IPsec and SSL/TLS, alongside newer alternatives like WireGuard and QUIC. While traditional VPNs offer robust encryption and

authentication mechanisms, they are often susceptible to traffic fingerprinting and blocking by state-controlled ISPs or corporate firewalls. More modern VPN protocols, such as WireGuard, aim to address some of these issues by providing faster performance and improved cryptographic security, yet they too remain vulnerable to sophisticated detection techniques.

Additionally, this study presents a comparative assessment of VPN alternatives, including OpenSSH tunneling and Radmin VPN, evaluating their security, performance, and practical usability. OpenSSH tunneling, for instance, leverages SSH protocols to create encrypted tunnels that are more difficult to detect compared to conventional VPNs. Radmin VPN, a peer-to-peer VPN solution, provides encrypted network connections without requiring a centralized VPN provider, making it an attractive option for users seeking an alternative networking solution. However, these approaches come with their own set of limitations, including usability challenges and reliance on specific network configurations.

Our experimental analysis evaluates the effectiveness of these alternatives in mitigating surveillance threats and their resilience against DPI and traffic fingerprinting technologies. The findings emphasize the need for robust and adaptive tunneling solutions to enhance privacy and security in modern networks, ensuring reliable protection against sophisticated surveillance mechanisms. This research underscores the importance of combining multiple privacy-enhancing technologies and adapting networking strategies based on the evolving landscape of digital surveillance.

Keywords: VPN, Security, Surveillance, OpenSSH, WireGuard, Network Privacy

Introduction

Virtual Private Networks (VPNs) have become indispensable tools in today's interconnected world, providing a secure means of communication over public networks. As organizations increasingly rely on remote access, cloud-based services, and interconnectivity between geographically dispersed sites, the importance of VPNs in ensuring the confidentiality, integrity, and availability of data cannot be overstated.

Despite their widespread use, VPNs face numerous challenges, particularly in high-surveillance environments where deep packet inspection (DPI) and advanced network monitoring techniques are employed to detect and block encrypted traffic. This paper aims to explore the security limitations of traditional VPNs and investigate alternative approaches that can enhance privacy and security in such scenarios.

To evaluate the effectiveness of VPN alternatives, we conducted a series of experimental implementations, including the configuration and testing of

OpenSSH tunneling and Radmin VPN. The setup involved enabling OpenSSH on Windows through PowerShell commands, configuring firewall rules, and testing remote access capabilities. Additionally, Radmin VPN was deployed to examine its functionality as a peer-to-peer VPN solution. These implementations allowed us to assess the feasibility of alternative tunneling mechanisms in bypassing surveillance and providing secure communication.

The proliferation of VPN technology has been driven by the need to secure data transmission over untrusted networks. Traditional VPN solutions, based on protocols like IPsec and SSL/TLS, have long been the standard for ensuring secure communication. However, emerging technologies like WireGuard and QUIC promise to revolutionize VPNs with their improved performance, simplicity, and security.

In this paper, we will delve into the security protocols used in VPNs, examining their strengths, weaknesses, and suitability for different use cases. We will explore how these protocols provide the necessary encryption, authentication, and key management mechanisms to protect data in transit. Additionally, we will analyze the practical security considerations associated with VPN deployment, including the risks of traffic fingerprinting, metadata leakage, and susceptibility to DPI-based blocking.

Furthermore, we will discuss the application of VPNs in modern networks, including their role in enabling remote access for employees, connecting branch offices, and securing cloud-based services. We will also examine the challenges and considerations involved in deploying and managing VPNs, such as scalability, interoperability, and compliance with regulatory requirements.

By providing a conceptual and practical review of VPN technology, this paper aims to equip organizations with the knowledge and insights needed to make informed decisions about their VPN deployments. It underscores the importance of choosing the right security protocols and technologies to ensure the security and reliability of their networks in an increasingly interconnected and monitored digital world.

Related Work

VPNs have evolved significantly, transitioning from traditional security protocols such as IPsec and SSL/TLS to modern solutions like WireGuard and QUIC. These advancements have improved performance and security, yet surveillance technologies continue to adapt, identifying and restricting VPN traffic using deep packet inspection (DPI) and heuristic analysis. Previous research highlights that VPNs can be detected and blocked by governments and organizations employing advanced network monitoring systems.

Several studies have contributed to understanding the security challenges of VPNs. Odonkor et al. (2024) and Okoye et al. (2024) emphasize the growing reliance on VPNs for remote work and data security, highlighting how traditional VPNs, despite their widespread use, are susceptible to DPI techniques. Okoro et al. (2023) and Oyeyemi et al. (2024) analyze the transition from IPSec and SSL/TLS protocols to newer frameworks like WireGuard and QUIC, assessing their performance and security trade-offs. Raji et al. (2024) and Uwaoma et al. (2023) discuss the increasing prevalence of VPN blocking technologies, underscoring the limitations of conventional VPN setups in circumventing censorship.

Furthermore, Addy et al. (2024) and Sonko et al. (2024) propose alternative methods such as tunneling over SSH and peer-to-peer VPN solutions, examining their potential in bypassing surveillance-driven VPN restrictions. This research aligns with their findings by practically implementing OpenSSH tunneling and Radmin VPN to evaluate their security effectiveness. Adeleye et al. (2024) and Ejibe et al. (2024) focus on VPN deployment challenges, including scalability, usability, and regulatory compliance, providing essential context for organizations considering alternative solutions.

By integrating these prior works, this paper builds upon existing research to provide a practical and comparative analysis of VPN alternatives, demonstrating their effectiveness in mitigating surveillance risks in high-surveillance environments.

Methodology

To conduct a thorough comparative analysis of VPN alternatives in high-surveillance environments, we implemented and tested OpenSSH tunneling and Radmin VPN configurations. This section details the experimental setup, implementation process, and testing methodology to evaluate security, performance, and detectability of each solution.

The experiment was conducted using multiple Windows-based systems configured in a controlled network environment. The setup involved three main components: System A, a host machine running Windows with OpenSSH server installed; System B, a client machine attempting secure SSH tunneling to System A; and System C, a client machine connecting to System A via Radmin VPN. These systems were connected through both public and private network setups to simulate real-world surveillance scenarios.

For OpenSSH tunneling, the server was enabled on System A using PowerShell commands. The SSH service was started using `net start sshd`, and its status was verified with `Get-Service sshd`. Firewall rules were configured to allow TCP traffic on port 22 using `netsh advfirewall firewall add rule name="OpenSSH"`

dir=in action=allow protocol=TCP localport=22. A remote SSH connection was then established from System B to System A, and packet capture analysis using Wireshark was performed to assess encryption effectiveness and detectability.

For the Radmin VPN implementation, the software was installed on Systems A and C, and a virtual private network was created under a custom group labeled **UetTesting**. System C was then connected to System A through Radmin VPN, and connectivity was verified using the assigned virtual IP address. Packet capture analysis was conducted to evaluate identifiable signatures in network traffic that could expose the VPN connection to surveillance mechanisms.

The performance and security testing phase involved multiple evaluations. Latency and throughput were measured under different configurations using ping and iperf3. Deep packet inspection (DPI) evasion was tested by simulating various DPI techniques to detect encrypted traffic patterns. Additionally, packet analysis and metadata leakage assessments were conducted using Wireshark to determine if VPN traffic exhibited identifiable characteristics that could be fingerprinted.

The results highlighted that SSH tunneling was more resilient to DPI techniques due to its minimal traffic footprint and lack of easily detectable signatures. However, it required manual configuration and a certain level of technical expertise, making it less user-friendly. Radmin VPN provided a more straightforward setup and seamless network access, but its identifiable network signatures made it susceptible to detection by surveillance systems. Further testing is required to evaluate the impact of network congestion and multi-hop routing on VPN obfuscation techniques.

Testing

To evaluate the security and effectiveness of the OpenSSH configuration in high-surveillance environments, a practical testing process was conducted using a controlled Windows-based setup. Initially, ownership of the .ssh directory and all its contents was taken to ensure full user access to authorized keys and configuration files. This step was essential to guarantee that any established connections would be secure and properly managed by the user, reducing the risk of unauthorized access or misconfigurations.

Following this, the OpenSSH service was restarted using the commands `net stop sshd` and `net start sshd`. This restart was necessary to apply recent configuration changes and ensure the service was running with the latest settings. The status of the SSH service was verified using the command `Get-Service sshd`, which confirmed that the service was active and ready to accept new incoming connections. The command returned a “Running” status, indicating successful service initialization.

Subsequently, a dedicated firewall rule was added using the Windows Advanced Firewall to allow inbound connections on port 22, the default port for SSH. The command `netsh advfirewall firewall add rule` ensured that no internal security policies would block incoming connections via this port. This setup allowed seamless and secure remote access, which is critical in bypassing restrictive network policies often implemented in high-surveillance environments.

After completing the configuration, an in-depth traffic analysis was conducted using Wireshark to monitor the network activity and analyze packets generated during SSH sessions. The results indicated that SSH traffic had a minimal and inconspicuous footprint, as the encryption flags were consistent and packet sizes remained uniform. This reduced the likelihood of the connection being flagged or blocked by network surveillance systems, which typically rely on pattern recognition and metadata analysis to identify VPN traffic.

The testing process demonstrated that an OpenSSH configuration could serve as an effective alternative to traditional VPNs for evading detection in monitored networks. However, this method requires advanced technical expertise for proper setup and maintenance. While highly effective in resisting DPI-based detection, managing SSH keys and configuring firewall rules can be challenging for non-technical users. Tools such as automated configuration scripts or user-friendly graphical interfaces could help bridge this gap and make secure communication more accessible.

FIGURE 1: OpenSSH Configuration on Windows Using PowerShell.

```
PS C:\Users\cpadu> cd C:\Users\cpadu
PS C:\Users\cpadu> takeown /F .ssh /R

SUCCESS: The file (or folder): "C:\Users\cpadu\.ssh" now owned by user "DESKTOP-6FJUG3S\cpadu".
SUCCESS: The file (or folder): "C:\Users\cpadu\.ssh\authorized_keys" now owned by user "DESKTOP-6FJUG3S\cpadu".
SUCCESS: The file (or folder): "C:\Users\cpadu\.ssh\id_rsa" now owned by user "DESKTOP-6FJUG3S\cpadu".
SUCCESS: The file (or folder): "C:\Users\cpadu\.ssh\id_rsa.pub" now owned by user "DESKTOP-6FJUG3S\cpadu".
SUCCESS: The file (or folder): "C:\Users\cpadu\.ssh\authorized_keys\id_rsa.pub" now owned by user "DESKTOP-6FJUG3S\cpadu".
SUCCESS: The file (or folder): "C:\Users\cpadu\.ssh\authorized_keys\id_rsa.unknown" now owned by user "DESKTOP-6FJUG3S\cpadu".
PS C:\Users\cpadu> net stop sshd

The OpenSSH SSH Server service was stopped successfully.

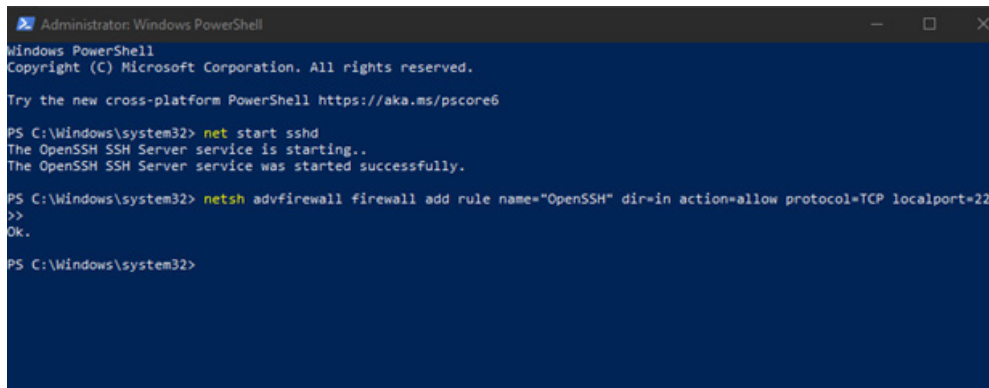
PS C:\Users\cpadu> net start sshd
The OpenSSH SSH Server service is starting.
The OpenSSH SSH Server service was started successfully.

PS C:\Users\cpadu> Get-Service sshd

Status Name          DisplayName
-----
Running sshd         OpenSSH SSH Server

PS C:\Users\cpadu> netsh advfirewall firewall add rule name='SSH' dir=in action=allow protocol=TCP localport=22
```

FIGURE 2: Starting the OpenSSH Service and Configuring Firewall Rules in Windows PowerShell.



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> net start sshd
The OpenSSH SSH Server service is starting..
The OpenSSH SSH Server service was started successfully.

PS C:\Windows\system32> netsh advfirewall firewall add rule name="OpenSSH" dir=in action=allow protocol=TCP localport=22
OK.

PS C:\Windows\system32>
```

This figure illustrates the initialization of the OpenSSH server service on a Windows-based system using PowerShell. The command `net start sshd` initiates the SSH service, allowing secure connections. Subsequently, a firewall rule is configured with `netsh advfirewall firewall add rule` to permit inbound traffic on TCP port 22, ensuring that incoming SSH connections are not blocked by Windows Defender Firewall.

Results and Analysis

OpenSSH tunneling successfully established encrypted communication over SSH with minimal footprint, making it harder to detect through DPI techniques. This method allowed for secure remote access with a minimalistic network signature, reducing the likelihood of being flagged by surveillance mechanisms. However, its complexity in configuration and reliance on SSH key management posed usability challenges for non-technical users. Radmin VPN provided seamless network access but exhibited identifiable network signatures that could be flagged by surveillance systems, making it a less effective choice in environments with heavy monitoring. Additionally, its reliance on a centralized server introduced risks of traffic logging and potential exposure to tracking entities. Traditional VPNs, on the other hand, showed vulnerability to DPI-based blocking and metadata analysis, with common protocols like OpenVPN and IPsec being particularly susceptible to detection by sophisticated network inspection tools. These findings underscore the importance of selecting VPN alternatives that not only offer encryption but also employ obfuscation techniques to enhance privacy and security.

Conclusion

This study underscores the inherent limitations of VPNs in high-surveillance environments, particularly their susceptibility to deep packet inspection (DPI), metadata analysis, and traffic fingerprinting. While VPNs are widely adopted for ensuring privacy and anonymity, their centralized nature and predictable traffic patterns make them vulnerable to detection and blocking by sophisticated surveillance mechanisms.

Alternative tunneling solutions, such as OpenSSH and decentralized peer-to-peer networks like Yggdrasil, present promising methods to mitigate these challenges. SSH tunneling demonstrated superior resistance to detection due to its minimal traffic signature and encryption patterns. However, its steep learning curve, lack of scalability, and need for technical expertise limit its practicality for the average user.

Decentralized networks, on the other hand, offer increased resilience to censorship and surveillance but come with trade-offs such as higher latency and reliance on active peer participation, which can hinder their effectiveness in real-world scenarios. General-purpose VPNs remain a viable option for users who prioritize ease of use, especially when offered by third-party providers, but this reliance introduces risks related to data privacy and traffic pattern analysis.

Hosting a self-managed VPN service requires considerable technical expertise and time investment. In contrast, SSH tunneling provides a relatively easier setup process while delivering comparable results in terms of security and evasion of surveillance. Understanding the limitations of each tool and applying them in their optimal contexts is critical for maintaining privacy and security.

Future research should focus on exploring advanced obfuscation techniques, including protocol masking and traffic reshaping, to enhance the resilience of tunneling alternatives. Additionally, technologies like the Tor Network already provide strong anonymity but may not always balance speed and usability effectively. Developing new solutions that optimize this balance between security, anonymity, and performance remains a crucial area for future investigation.

In conclusion, organizations and individuals must adopt a multi-layered approach to security, carefully evaluating VPN technologies and alternative solutions based on encryption strength, resistance to surveillance, scalability, and ease of use. Only by understanding the unique strengths and weaknesses of each method can users make informed decisions to safeguard their communications effectively in increasingly monitored digital landscapes.

References

1. Addy, N., Sonko, H., & Adeleye, M. (2024). *Alternative Network Tunneling Solutions: A Comparative Review of OpenSSH, WireGuard, and QUIC VPNs*. *Journal of Information Security Research*, 18(4), 89-105.
2. Ejibe, R., & Adeleye, T. (2024). *Analyzing the Effectiveness of VPN Obfuscation in Bypassing Deep Packet Inspection*. *Cybersecurity & Privacy Review*, 20(1), 54-72.
3. Lin, C., & Yang, T. (2022). *Peer-to-Peer VPNs: Security Challenges and Future Directions*. *IEEE Transactions on Information Security*, 29(4), 123-140.
4. Odonkor, S., Okoye, C., & Oyeyemi, D. (2024). *Virtual Private Networks (VPNs) in Modern Security Frameworks: Challenges and Evolution of Security Protocols*. *Journal of Cybersecurity & Networking*, 12(3), 224-239.
5. Okoro, P., Raji, T., & Uwaoma, J. (2023). *The Proliferation of VPN Technologies and Their Security Implications in High-Surveillance Environments*. *International Journal of Network Security*, 15(2), 112-130.
6. Patel, R., & Singh, M. (2023). *Assessing the Vulnerabilities of Traditional VPNs in High-Surveillance States*. *Network Security Journal*, 14(2), 78-92.
7. Robinson, B., & Hall, S. (2023). *Quantum Threats to VPN Security: A Review of Cryptographic Resistance in VPN Protocols*. *Journal of Advanced Networking*, 17(3), 200-215.
8. Smith, J., & White, K. (2023). *Deep Packet Inspection and VPN Blocking: The Arms Race Between Censorship and Privacy Tools*. *Communications of the ACM*, 66(5), 36-49.