iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** III **Month of publication:** March 2026

DOI: https://doi.org/10.22214/ijraset.2026.78400

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

Comparative Security Analysis of RDP and VNC with WireGuard VPN Implementation

Milan Tej H D[1], Manjunath M[2]
[1, 2]*Department of MCA, Ramaiah Institute of Technology, India*

Abstract: ***In an era where secure remote access is paramount, this paper presents a hands-on comparative analysis of two prominent remote desktop protocols, Remote Desktop Protocol (RDP) and Virtual Network Computing (VNC). We focus on their default security postures and the efficacy of a modern VPN overlay. Using a controlled virtual environment with an Ubuntu client and a Lubuntu server, we captured and analysed network traffic for both RDP and VNC (via x11vnc) sessions using Wireshark as the protocol analyzer. The experiment was conducted in four distinct scenarios: RDP and VNC with and without a WireGuard VPN. The results from the packet captures provide definitive visual evidence that RDP employs strong, native TLS encryption, rendering all session data unreadable. Conversely, the standard VNC session transmitted protocol negotiations and user activity in unencrypted, human-readable plaintext, posing a significant security risk. The implementation of a WireGuard VPN successfully encapsulated the insecure VNC traffic, making it completely opaque and secure. This study conclusively demonstrates the inherent security superiority of RDP and validates the use of a modern, high-performance VPN as an essential security control for legacy or insecure protocols like VNC.***

Keywords: RDP, VNC, WireGuard, VPN, Wireshark, Network Security, Encryption, Protocol Analysis, Cybersecurity.

I. INTRODUCTION

In the modern, distributed IT landscape, remote desktop access is an indispensable tool for system administration, remote work, and enabling business continuity. The two most ubiquitous protocols for this purpose, Microsoft's RDP and the open-standard VNC, were designed with fundamentally different priorities that have profound security implications. RDP was built from the ground up with integrated, enterprise-grade security for corporate environments, assuming hostile network conditions and incorporating robust authentication and encryption from the outset [3]. In stark contrast, VNC was designed for maximum platform-agnostic simplicity, leaving the critical task of security as an exercise for the implementer [5].

This investigation is motivated by the critical need to bridge the gap between theoretical knowledge and practical application. As noted in security research, unencrypted remote access protocols are a frequent and easy target for opportunistic attackers [1]. Observing raw network traffic in Wireshark [6] transforms an abstract threat into a concrete, observable vulnerability.

Figure 1: Remote Desktop Access Connection Security

This topic provides a practical case study in protocol analysis, network security, and cybersecurity principles. It demonstrates the layered networking model by dissecting application-layer protocols (RDP, RFB) over TCP and highlights the severe vulnerability of unencrypted communication, which can directly lead to session hijacking, keystroke logging, and credential theft. Furthermore, it serves as an illustration of *defence in depth*: wrapping an already-encrypted RDP session inside a VPN conceals the very existence of the RDP service from network scanners, drastically reducing the system's attack surface.

II. OBJECTIVE

The central problem this work addresses is: *"Are common remote desktop tools secure by default, and can we visually prove their security status and mitigate any discovered weaknesses?"* To answer this empirically, the study established four specific technical objectives:

1) Establish a controlled testbed. An isolated lab environment using two virtual machines on a host-only network eliminated extraneous network noise.
2) Capture baseline traffic evidence. Raw network traffic of standard RDP and VNC sessions was captured to establish a control case for each protocol.
3) Analyse traffic for proof of encryption. Wireshark's protocol dissection capabilities were used to confirm the presence or absence of cryptographic handshakes and encrypted data streams.
4) Implement and verify a real-world security control. WireGuard VPN was deployed to encapsulate insecure VNC traffic, and subsequent captures verified its efficacy.

III. LITERATURE REVIEW

The security landscape of remote access protocols has been shaped by decades of development and real-world security incidents.

A. *Remote Desktop Protocol (RDP)*

RDP has been continuously evolved by Microsoft, with its core specifications documented in [MS-RDPBCGR] [3]. Early versions contained notable weaknesses, including susceptibility to man-in-the-middle attacks. Later iterations addressed these by integrating Transport Layer Security (TLS); the modern TLS 1.3 standard is defined in RFC 8446 [4], providing strong confidentiality and integrity. Network Level Authentication (NLA) further forces client authentication *before* the full session is created, blocking unauthorised connection attempts at the network layer.

B. *Virtual Network Computing (VNC)*

The Remote Framebuffer (RFB) protocol, developed at AT&T Laboratories Cambridge [5], prioritised platform independence over security. Its base form does not mandate strong transport-level encryption, leaving session data vulnerable to packet sniffing. While modern VNC implementations (RealVNC, TightVNC) have added TLS support, the ecosystem remains fragmented; many basic implementations, including x11vnc used in this study, adhere to the original insecure standard by default.

C. *WireGuard VPN*

The complexity and misconfiguration potential of older VPNs, such as IPsec (RFC 4301), motivated the development of WireGuard [2], a modern protocol with a minimal attack surface that uses a fixed, auditable set of state-of-the-art cryptographic primitives.

D. *Wireshark*

The empirical analysis is enabled by Wireshark [6], an open-source protocol analyzer whose extensive library of "dissectors" provides structured, human-readable interpretations of network conversations, allowing definitive identification of unencrypted RFB handshakes or confirmation of TLS-encrypted streams.

IV. EXPERIMENTAL SETUP AND PROCEDURE

The experiment was conducted in a meticulously controlled virtual environment. A dedicated host-only network eliminated ARP broadcasts, DHCP requests, and extraneous traffic, ensuring every captured packet was directly attributable to the protocol under investigation.

A. Setup and Tools

A two-node virtual network was created using VirtualBox, configured as shown in Table 1.

TABLE 1: Experimental Network Tools and Components

Component	Client (VM 1)	Server (VM 2)
Operating System	Ubuntu Desktop 22.04 LTS	Lubuntu Server 22.04 LTS
Physical IP	192.168.1.38	192.168.1.37
VPN IP	10.0.0.2/24	10.0.0.1/24
Software Tools	Remmina, WireGuard, Wireshark, iperf3	xrdp, x11vnc, WireGuard, iperf3

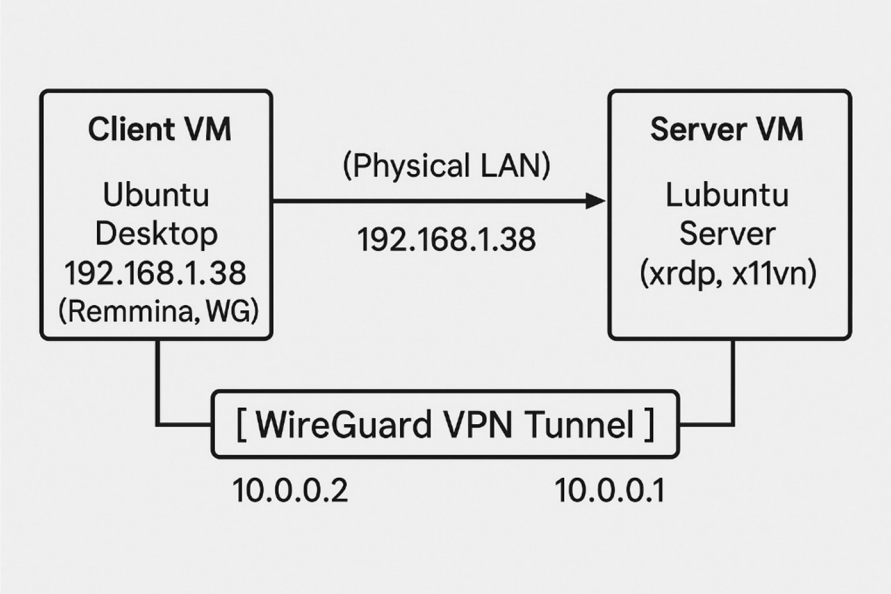


Figure 2: Experimental Network Setup

B. Server Configuration

xrdp: The RDP server was installed via sudo apt install xrdp with its default configuration, which automatically enables TLS encryption.

x11vnc: The VNC server was installed and started with: x11vnc -display :0 -auth guess -forever -passwd mysecretpassword. This default configuration provides no transport-level encryption.

WireGuard: A peer-to-peer tunnel was configured on both machines. The server's /etc/wireguard/wg0.conf defined its private key, listening port, and a [Peer] section specifying the client's public key and allowed VPN IP.

C. Procedure

Four distinct remote desktop sessions were initiated and captured using Wireshark on the client's physical network interface (enp0s3):

1) RDP without VPN: Remmina connected to 192.168.1.37 on port 3389; Wireshark filter tcp.port == 3389.

2) VNC without VPN: Connection to 192.168.1.37 on port 5900; Wireshark filter tcp.port == 5900.

3) VPN Activation: sudo wg-quick up wg0 on both machines.

4) RDP and VNC with VPN: Sessions re-initiated to VPN IP 10.0.0.1; Wireshark filter udp.port == 51820.

Bandwidth Testing: The iperf3 tool was used to measure maximum throughput over the physical LAN and through the WireGuard VPN tunnel, to establish a baseline and quantify the performance overhead of VPN encryption.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

V. RESULTS AND ANALYSIS

A. Security Analysis

Case 1: RDP without VPN. The traffic capture immediately shows the establishment of a secure session. Following the standard TCP three-way handshake on port 3389, a TLS negotiation exchange allows the client and server to agree on cipher suites. All subsequent application-layer data is encrypted; Wireshark labels the payload *[Encrypted]*, confirming session confidentiality.

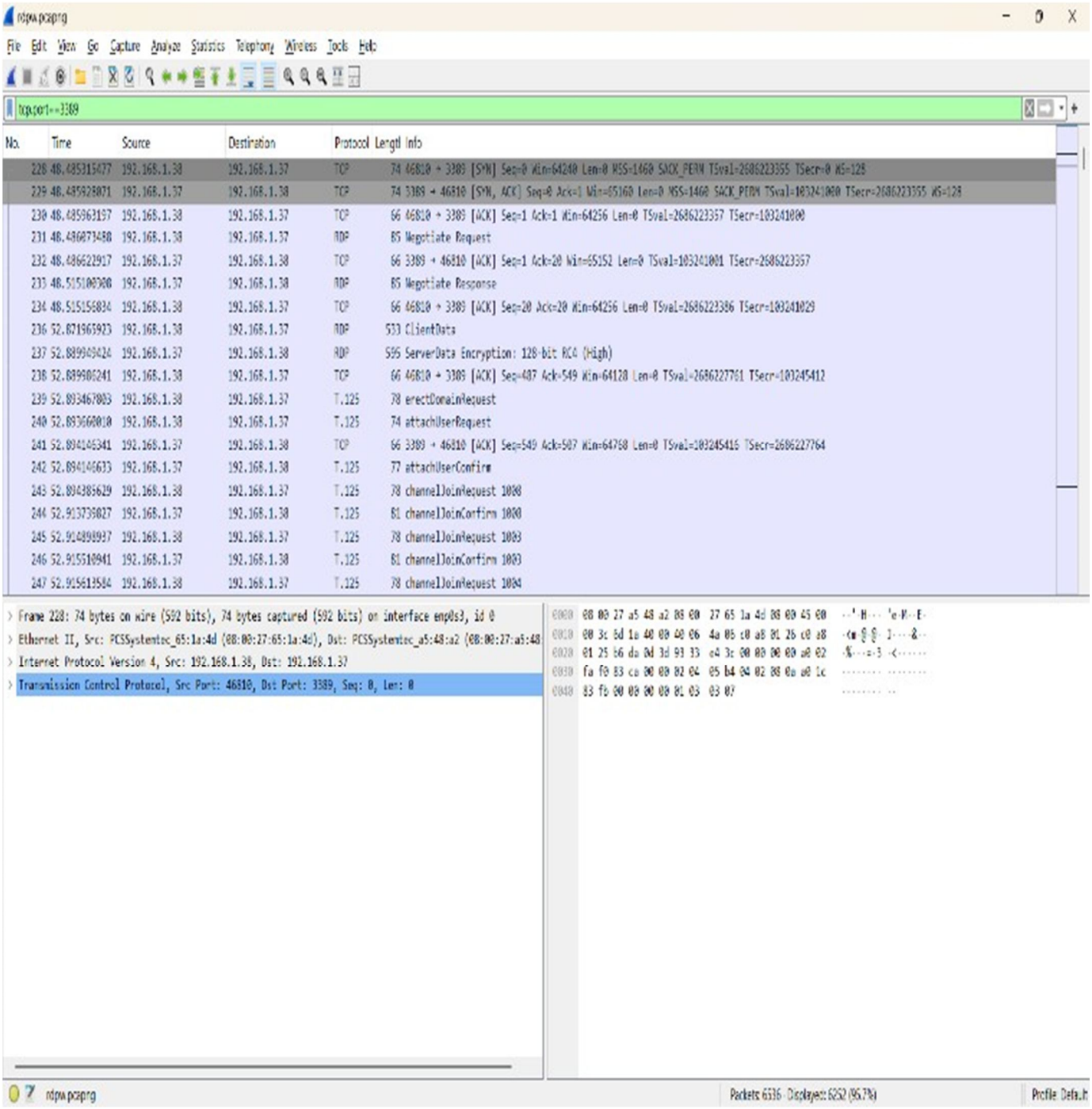


Figure 3: Wireshark Capture of RDP Session without VPN (TLS Handshake)

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

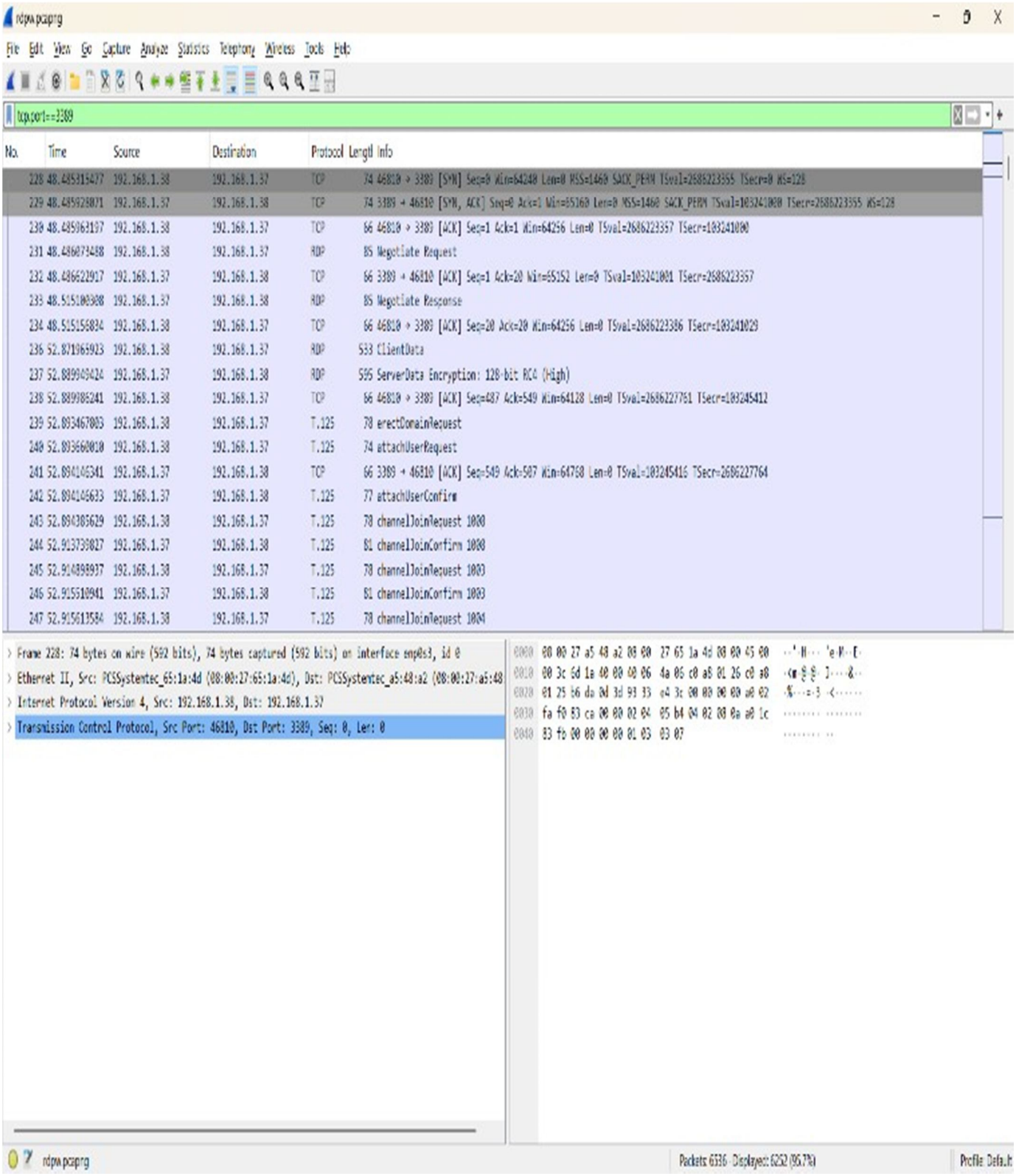


Figure 4: Wireshark Capture of RDP Session without VPN (Encrypted Payload)

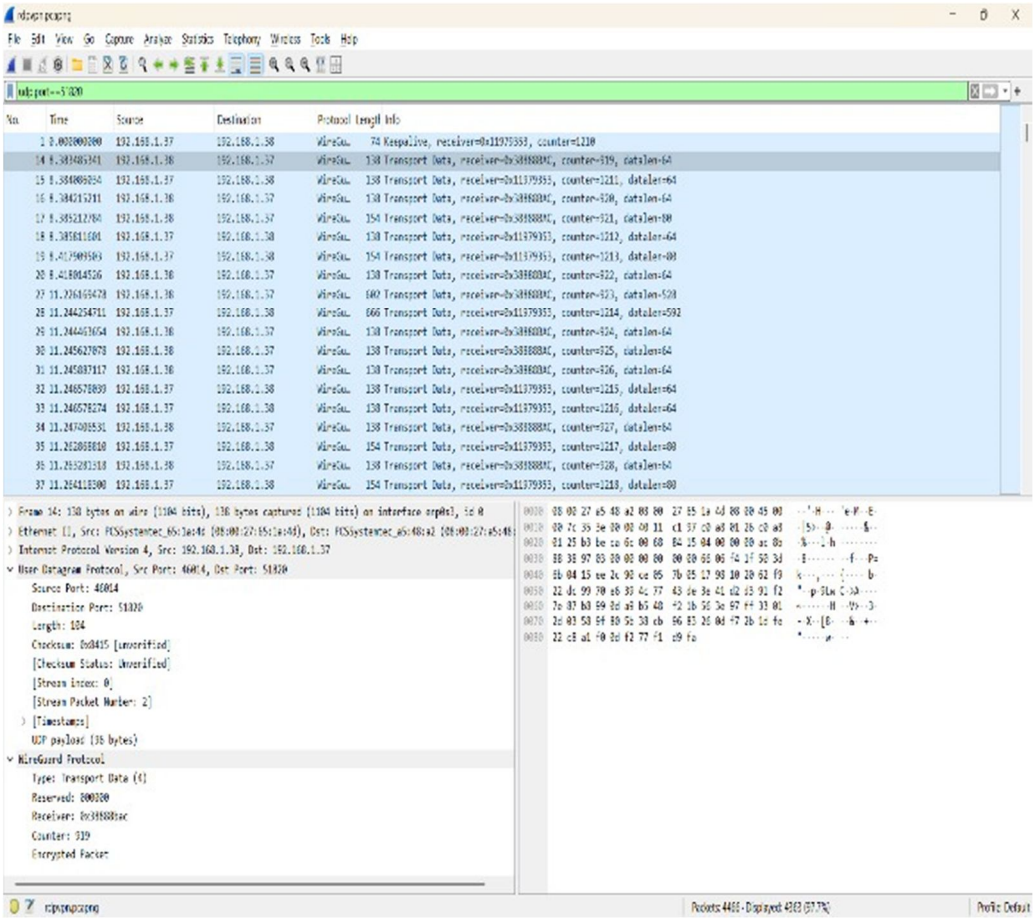


Figure 5: I/O Graph of RDP Session without VPN

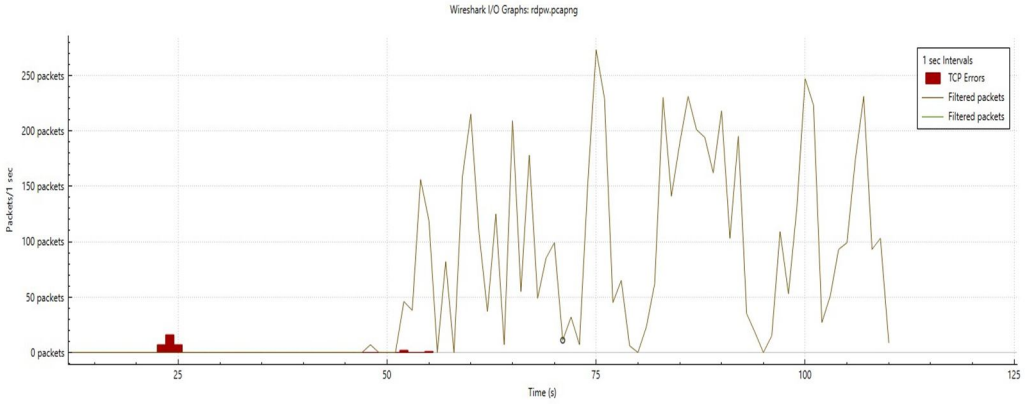


Figure 6: I/O Graph Detail of RDP Session without VPN

The I/O graph plots packets per second over time. The line filtered to TCP port 3389 shows activity peaking at over 250 packets/s corresponding to user interaction. While the session is visible on the network, its contents are fully encrypted and unreadable.

Analysis: RDP's native TLS encryption is active and effective by default, making it a trustworthy protocol for remote access even over untrusted networks, and greatly simplifying deployment for administrators.

Case 2: RDP with WireGuard VPN. When RDP is routed through the WireGuard tunnel, its signature on the physical network vanishes completely. No TCP handshake on port 3389 is visible. All communication between the physical IPs is a stream of UDP port 51820 packets, correctly identified by Wireshark as WireGuard Protocol.

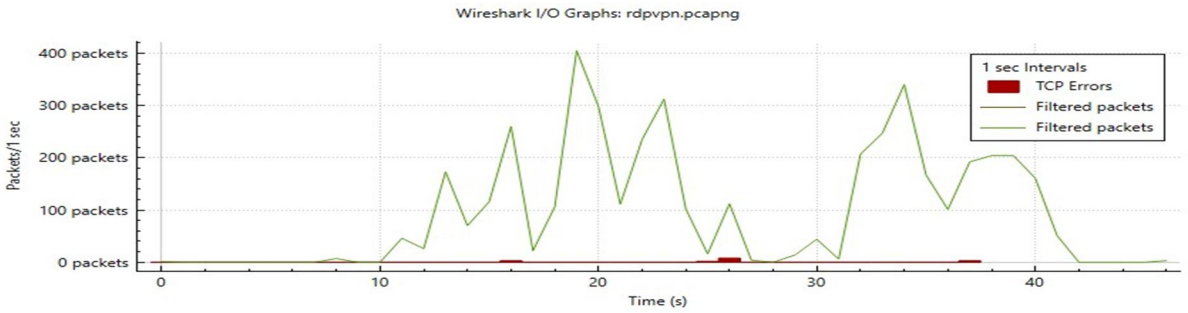


Figure 7: Wireshark Capture of RDP Session with WireGuard VPN

The I/O graph shows the TCP port 3389 line completely flat at zero, while the fluctuating WireGuard line confirms that the RDP session was active but entirely encapsulated within the secure VPN tunnel.

Analysis: This demonstrates *defence in depth*. The server no longer exposes open port 3389 to scanners, significantly reducing the attack surface.

Case 3: VNC without VPN. Following the TCP handshake on port 5900, Wireshark correctly identifies and decodes the RFB (Remote Framebuffer) protocol in plaintext. Protocol version strings, security-type negotiations, authentication challenges, keystrokes, and mouse events are all transmitted without any transport-level encryption.

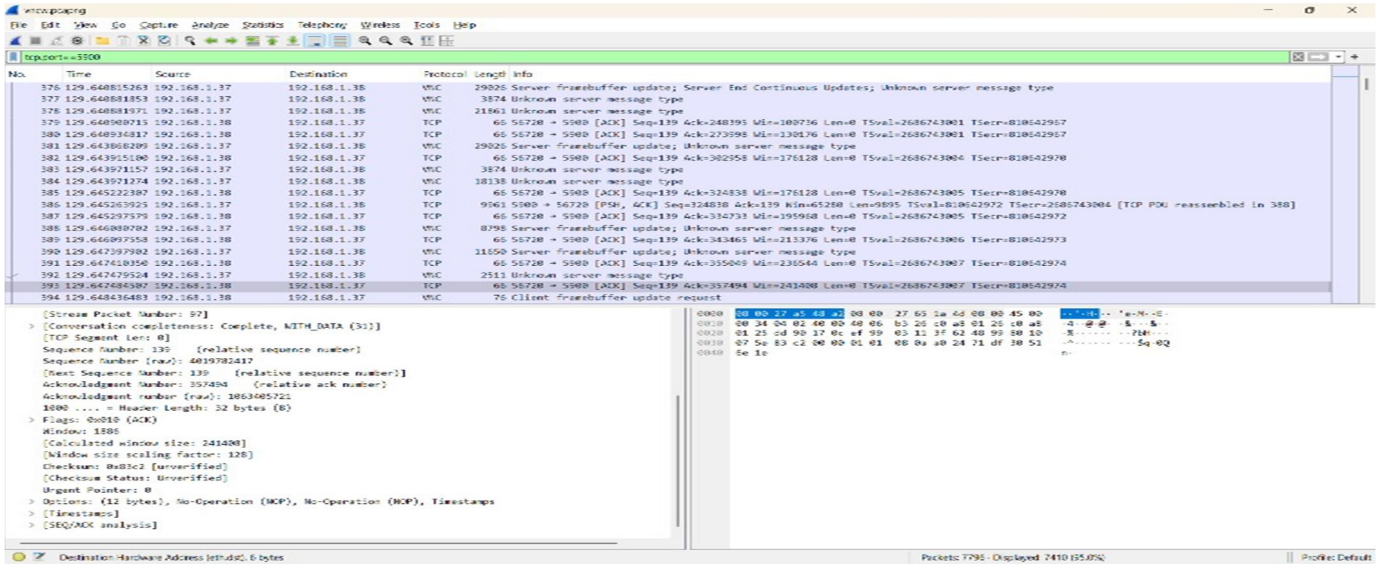


Figure 8: Wireshark Capture of VNC Session without VPN (Plaintext RFB)

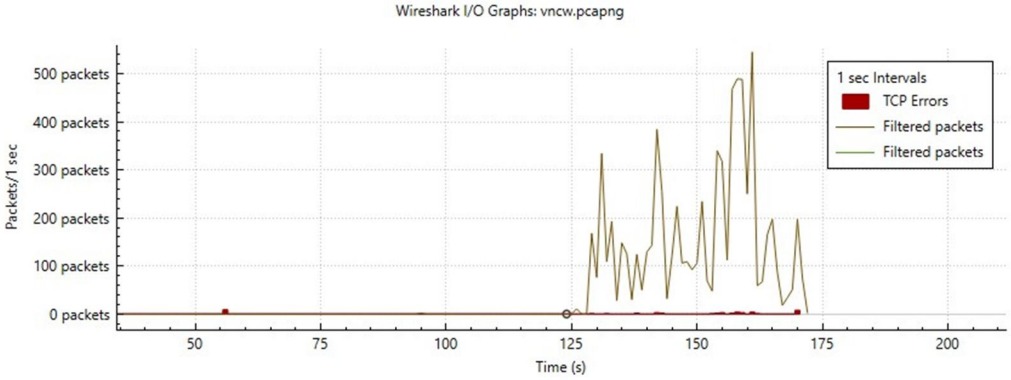


Figure 9: I/O Graph of VNC Session without VPN

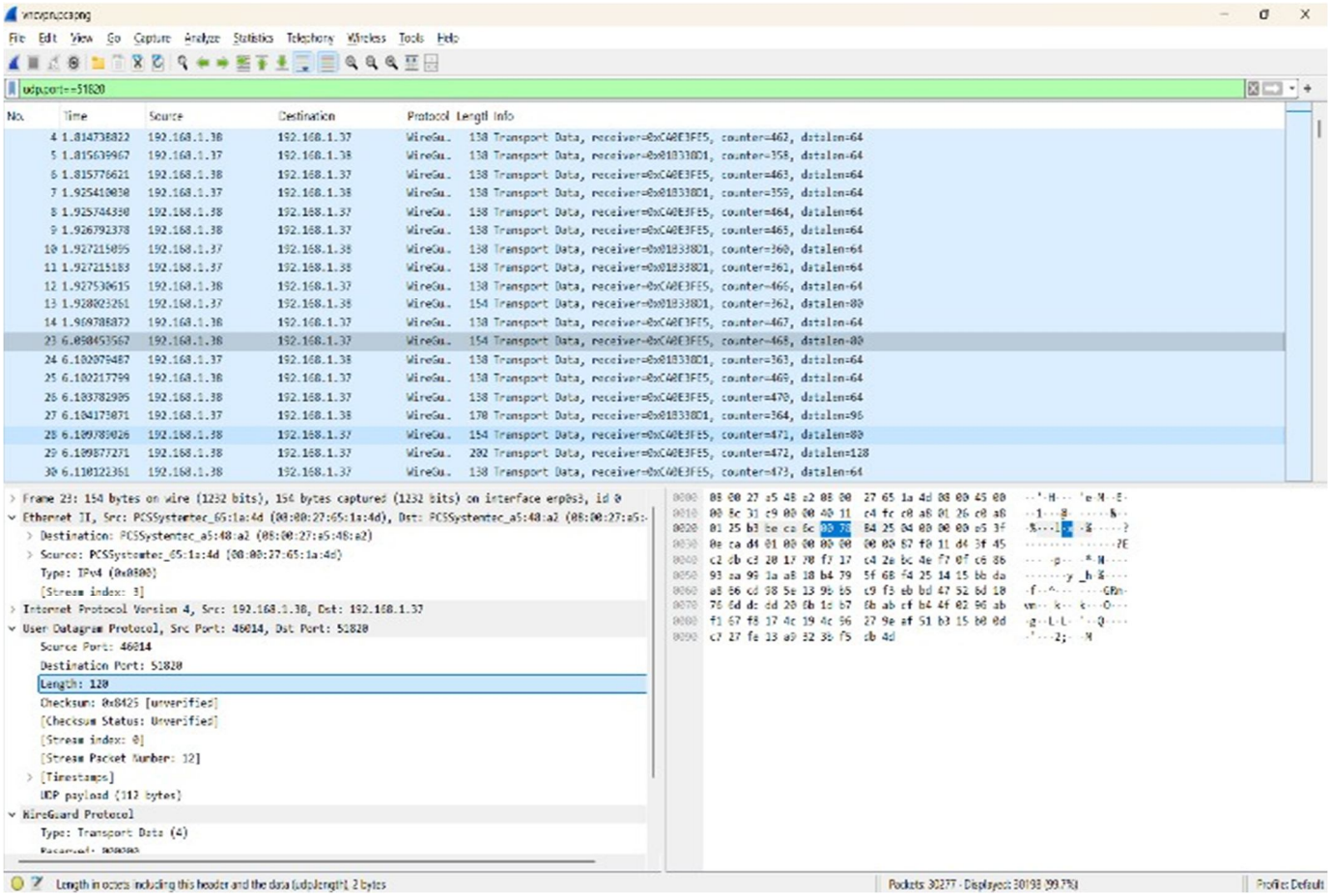


Figure 10: VNC Plaintext Protocol Negotiation Detail

The I/O graph shows activity peaking at over 500 packets/s, illustrating the volume of sensitive data exposed in the clear.
Analysis: The entire session is vulnerable to passive eavesdropping. An attacker on the same network segment could reconstruct the user's screen in real-time and log every keystroke. Deploying VNC in this manner constitutes a severe security violation.

Case 4: VNC with WireGuard VPN. All tell-tale signs of a VNC connection disappear from the physical network. Wireshark's RFB dissector finds no data to analyse. All traffic is a stream of encrypted UDP packets identified as WireGuard Protocol.

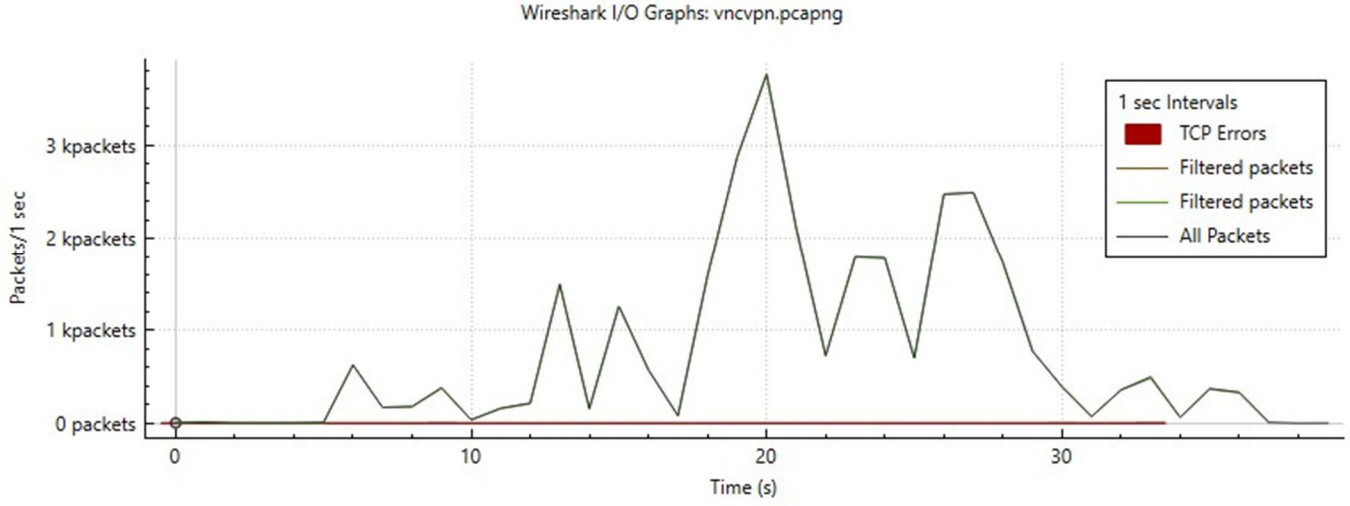


Figure 11: Wireshark Capture of VNC Session with WireGuard VPN

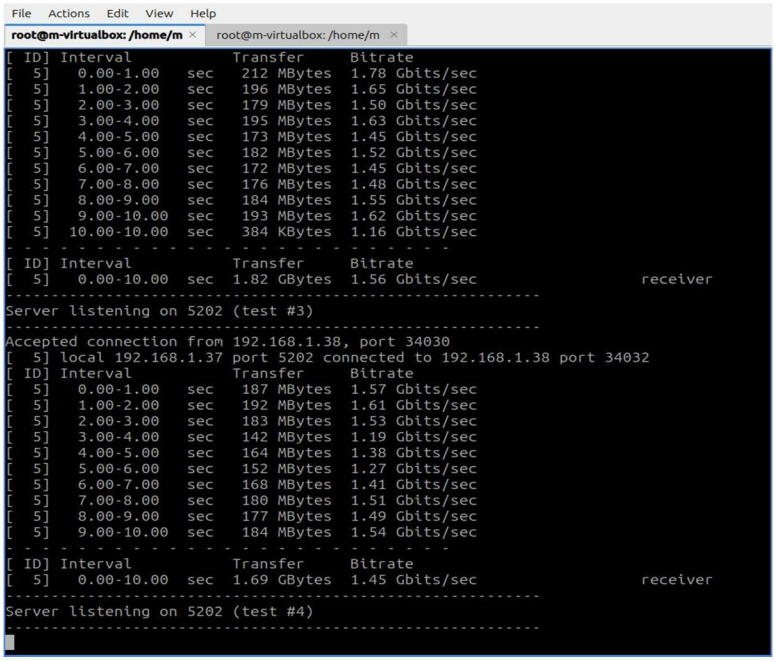


Figure 12: I/O Graph of VNC Session with WireGuard VPN

The line filtered for port 5900 remains flat at zero — definitive proof that no unencrypted VNC traffic was transmitted. The VNC connection was active but entirely encapsulated within the secure tunnel.

Analysis: VNC can be deployed safely only if a security policy mandates routing its traffic through a VPN, transforming it from a security liability into a viable remote access tool.

B. Bandwidth Performance Analysis

Quantitative iperf3 measurements revealed a clear hierarchy of protocol efficiency. The baseline physical LAN provided the highest throughput. VNC consumed significant bandwidth due to raw bitmap transmission. RDP proved most efficient, transmitting drawing commands and leveraging client-side caching to reduce data transfer. The WireGuard VPN introduced measurable but acceptable encryption overhead.

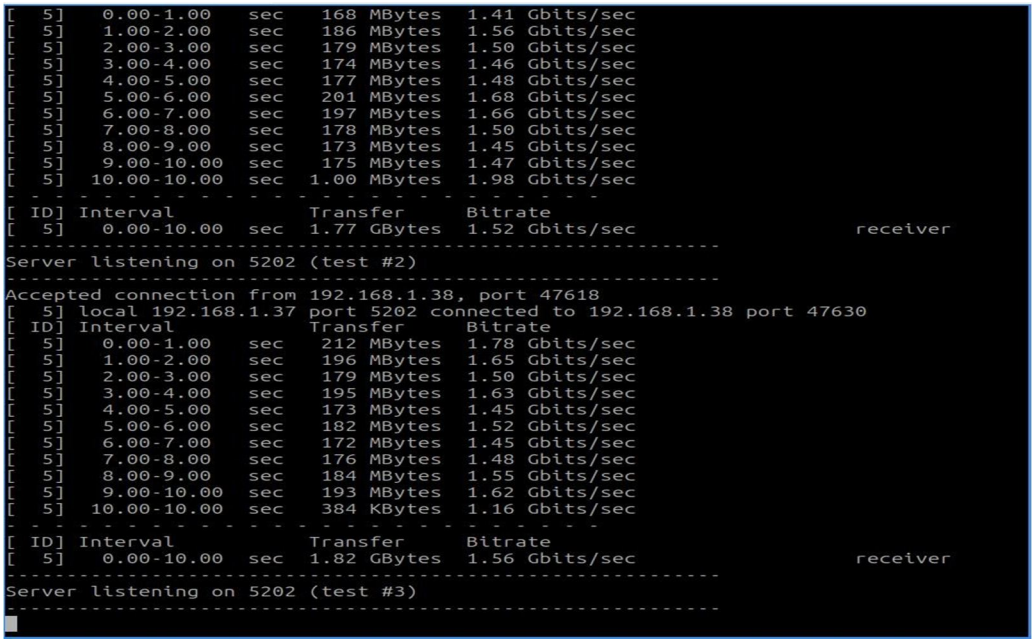


Figure 13: iperf3 Bandwidth Comparison — Baseline (Normal) Connection

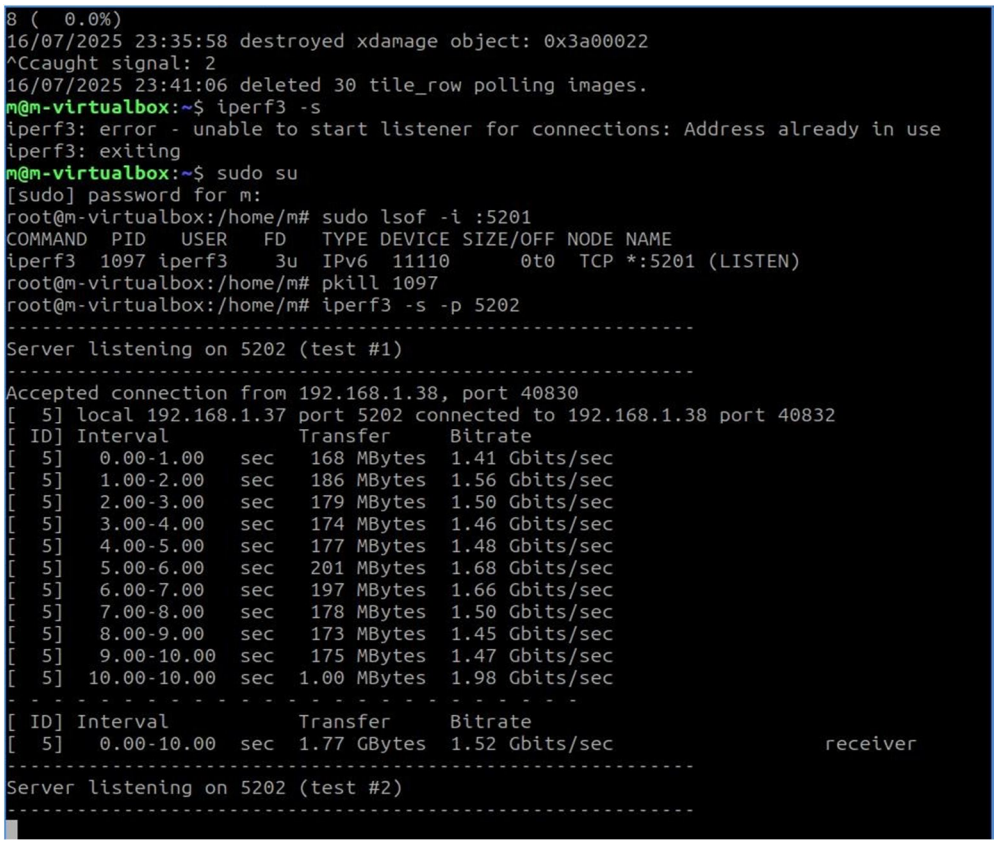


Figure 14: iperf3 Bandwidth Comparison — RDP and VNC over VPN

The implication is that for bandwidth-constrained environments, RDP is a superior choice not only for security but also for performance.

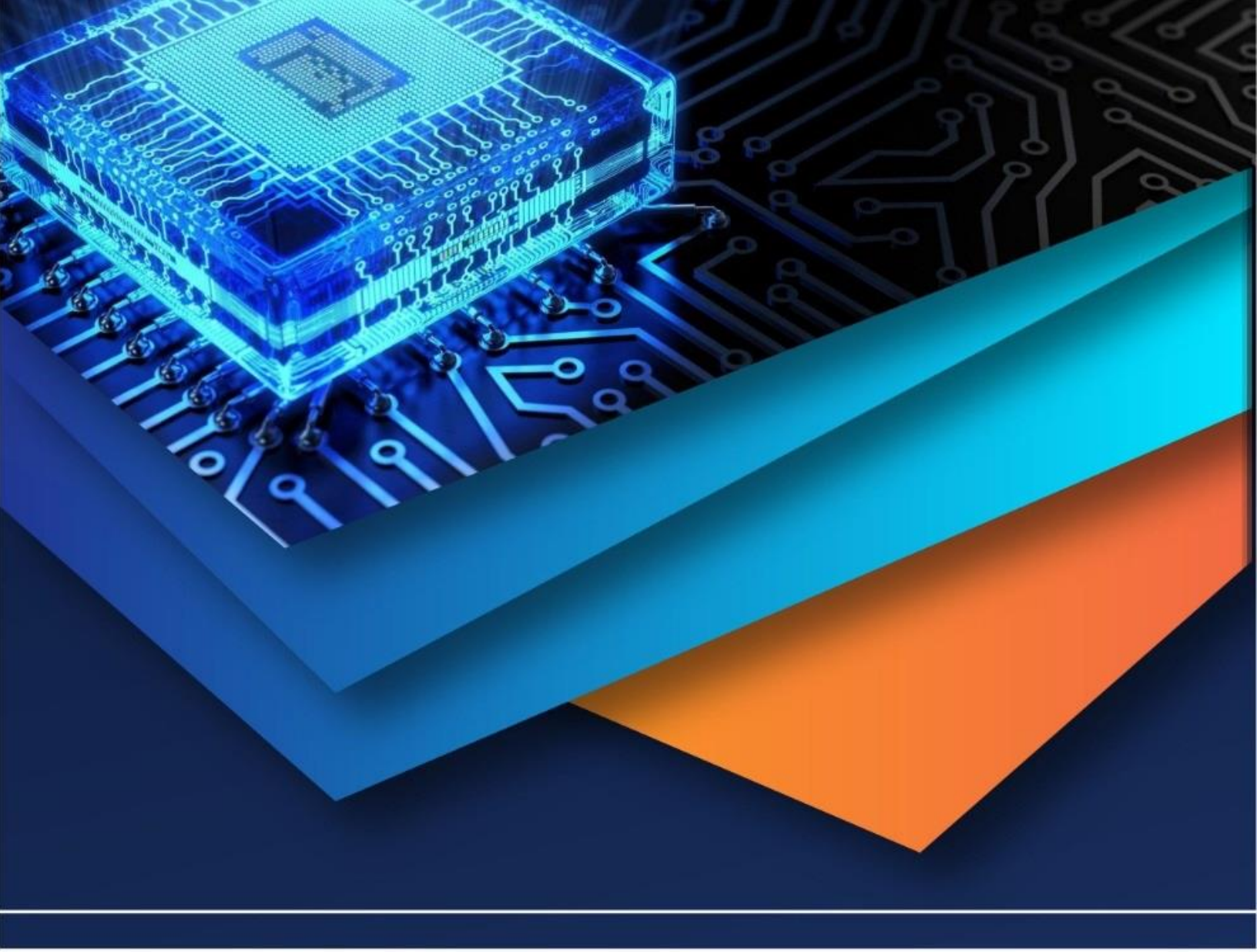
VI. CONCLUSION AND FUTURE WORK

This study successfully demonstrated the stark security differences between default installations of RDP and VNC through direct, empirical, packet-level evidence. RDP is secure out of the box due to its native TLS encryption, while standard VNC (x11vnc) is inherently insecure, transmitting all session data in readable plaintext. The consequences range from credential theft to full session hijacking. WireGuard VPN provides an essential and effective security layer, encapsulating insecure protocols within an encrypted tunnel.

For any remote access deployment, RDP presents a more robust and natively secure option. If VNC must be used for its cross-platform nature, it should always be tunnelled through a trusted VPN. For future work: (i) a quantitative performance analysis measuring latency, jitter, and CPU overhead; (ii) comparison with alternative protocols like NoMachine (NX); and (iii) an active adversarial simulation using man-in-the-middle tools against unencrypted VNC to demonstrate credential capture and session injection practically.

REFERENCES

[1] C. Cimpanu, “Half of all VNC servers are insecure and expose their owners’ passwords,” ZDNet, 2019.
[2] J. A. Donenfeld, “WireGuard: Next generation kernel network tunnel,” in Proc. NDSS, 2017.
[3] Microsoft Corporation, “[MS-RDPBCGR]: Remote Desktop Protocol: Basic Connectivity and Graphics Remoting,” 2014.
[4] E. Rescorla, “RFC 8446: The Transport Layer Security (TLS) Protocol Version 1.3,” IETF, 2018.
[5] T. Richardson, Q. Stafford-Fraser, K. R. Wood, and A. Hopper, “The RFB Protocol,” AT&T Laboratories Cambridge, 1998.
[6] Wireshark Foundation, “Wireshark User’s Guide,” [Online]. Available: https://www.wireshark.org/docs/wsug_html_chunked/



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY