# Detection of Screen Shadowing-Based Visual Data Exfiltration Attacks In VNC Systems

**Fejisha Dev E B[1], Suba A[2]**
[1] Dept of Electronics and Communication Engineering
[2]Assist prof, Dept of Computer Science Engineering
[1, 2] Francis Xavier Engineering College Tirunelveli - 627 003, Tamil Nadu, India

***Abstract-*** *With the increasing adoption of remote desktop technologies in enterprise environments, the security of Visual Network Computing (VNC) systems has become a critical concern. Screen shadowing attacks represent a significant threat vector where malicious actors silently capture sensitive visual data displayed on remote desktops, including passwords, financial information, and confidential documents. This paper proposes the design and implementation of a real-time detection system for identifying screen shadowing-based visual data exfiltration attacks in VNC environments. The proposed system incorporates a virtual laboratory environment consisting of Ubuntu-based VNC server and Kali Linux-based attacker systems connected through an isolated internal network. Network traffic analysis is performed using Wireshark and tshark tools to establish baseline traffic patterns during normal VNC usage. Attack simulations including rapid screen capture and high-quality stream extraction are conducted to generate attack traffic signatures. A Python-based detection engine utilizing threshold-based anomaly detection and machine learning algorithms, specifically Isolation Forest, is implemented to identify deviations from normal traffic patterns. The system provides real-time alerting capabilities and automated evidence capture for forensic analysis. Experimental results demonstrate that the proposed system achieves high detection accuracy with minimal false positives, effectively identifying screen shadowing attacks through bandwidth analysis, packet rate monitoring, and statistical pattern recognition. The proposed architecture provides organizations with an effective tool for protecting sensitive visual data in VNC-enabled remote work environments.*

***Keywords:*** VNC Security, Screen Shadowing, Visual Data Exfiltration, Anomaly Detection, Network Traffic Analysis.

## I. INTRODUCTION

Remote desktop technologies have become essential infrastructure components in modern enterprise environments, enabling remote work, technical support, and system administration across geographically distributed networks. Virtual Network Computing (VNC) is one of the most widely deployed remote desktop protocols, providing platform-independent graphical desktop sharing capabilities based on the Remote Framebuffer (RFB) protocol. The VNC architecture operates on a client-server model where the VNC server shares its screen content with connected VNC viewers, enabling real-time visual interaction with remote systems.The widespread adoption of VNC technology has been accelerated by the global shift toward remote work environments, particularly following the COVID-19 pandemic. Organizations across healthcare, finance, government, and corporate sectors increasingly rely on VNC connections to access sensitive systems and data remotely. However, this increased dependency on remote desktop technologies has introduced significant security vulnerabilities that traditional security mechanisms fail to address adequately.Screen shadowing attacks represent a particularly insidious threat to VNC security. In these attacks, malicious actors exploit legitimate VNC connections to silently capture and exfiltrate visual data displayed on remote desktops. Unlike traditional network intrusion attacks that target data in transit or at rest, screen shadowing attacks target data at the point of display, capturing sensitive information as it appears on screen. This includes login credentials being typed, financial documents being reviewed, confidential emails being read, and proprietary information being accessed.

## II. PROBLEM STATEMENT

The security of remote desktop technologies presents multiple challenges that current monitoring systems fail to address effectively. The primary challenge lies in the nature of visual data exfiltration, which occurs through legitimate protocol channels and does not generate traditional attack signatures. Organizations face significant difficulties in distinguishing between normal VNC usage and malicious screen capture activities.The first major challenge involves the lack of visibility into VNC traffic patterns. Conventional network monitoring tools treat VNC connections as encrypted data streams without understanding the visual content being transmitted. Security teams lack the ability to determine whether a VNC session involves normal administrative activities or aggressive screen capture operations designed to

exfiltrate sensitive visual data.The second challenge relates to insider threat detection. Screen shadowing attacks are often conducted by individuals with legitimate access credentials, including compromised accounts, malicious insiders, or attackers who have obtained valid credentials through phishing or social engineering. Traditional perimeter-based security controls are ineffective against these threats because the attacks originate from authenticated sessions.

## III. LITERATURE REVIEW

Remote desktop security has been an active area of research, with numerous studies addressing various aspects of VNC vulnerability assessment, attack characterization, and detection mechanism development. Richardson and Levine conducted foundational research on VNC protocol security, identifying multiple vulnerability classes including authentication weaknesses, encryption limitations, and session hijacking possibilities, revealing that standard VNC implementations transmit visual data with minimal protection against eavesdropping and replay attacks. The RFB protocol specification defines the client-server communication model used by VNC implementations, and security researchers have identified protocol-level weaknesses including predictable session identifiers, weak challenge-response authentication, and lack of forward secrecy. Research on visual data exfiltration has examined multiple attack vectors through which sensitive information displayed on screens can be captured and extracted, with studies by Zhou et al. demonstrating that screen capture malware represents a significant threat to data confidentiality, capable of extracting sensitive information that never exists in traditional data storage formats. Screen shadowing attacks specifically targeting remote desktop protocols were analyzed by Kumar and Sharma, who developed attack taxonomies categorizing different exfiltration techniques based on attack methodology and data capture mechanisms, identifying rapid screen refresh attacks, high-quality capture attacks, and selective region capture attacks as primary threat categories. Network traffic analysis has been extensively applied to security monitoring across various protocols and applications, with research by Paxson establishing foundational principles for network-based intrusion detection, demonstrating that traffic patterns carry signatures indicative of malicious activity.

## IV. PROPOSED METHODOLOGY

The proposed system utilizes a multi-stage threat intelligence pipeline that monitors VNC traffic, establishes behavioral baselines, and identifies anomalous patterns indicative of screen shadowing attacks through real-time analysis. The system begins with the deployment of a controlled virtual laboratory environment consisting of two virtual machines: an Ubuntu-based system configured as the VNC server representing the victim machine, and a Kali Linux-based system configured as the VNC client representing the attacker machine. Both virtual machines are connected through an isolated internal network using VirtualBox, ensuring controlled traffic capture and analysis without interference from external network activities. The VNC server is configured using TigerVNC standalone server listening on port 5901, with password authentication enabled and a lightweight desktop environment comprising Openbox window manager and xterm terminal for minimal resource consumption. Network traffic between the VNC server and client is captured using Wireshark and tshark tools, with specific filters applied to isolate VNC communication on the designated port. During the baseline establishment phase, normal VNC usage patterns are recorded through simulated administrative activities including file browsing, text editing, and application navigation, with traffic metrics including packet rates, bandwidth utilization, packet size distributions, and inter-arrival times computed and stored as baseline reference values.

## V. SYSTEM ARCHITECTURE

The architecture of the proposed system is designed to capture, analyze, and respond to screen shadowing attacks through an integrated monitoring pipeline. The architecture consists of four primary components: the victim environment, the attacker environment, the monitoring infrastructure, and the detection engine.

### 5.1 Overall System Architecture

**Figure 1 illustrates the complete system architecture showing the relationship between system components and data flow paths.**

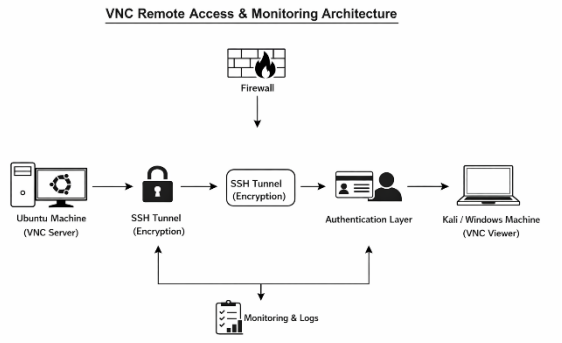


Fig 1: System Architecture of the Proposed VNC Attack Detection System

### 5.2 Victim Environment (VNC Server)

The victim environment consists of an Ubuntu virtual machine configured as a VNC server using TigerVNC. The system simulates a typical enterprise workstation containing sensitive data including documents, credentials, and confidential information. The VNC server is configured to accept connections on standard port 5901, with password authentication enabled.

Key components include:Ubuntu 22.04 LTS operating system, TigerVNC standalone server, Simulated sensitive data files, Network interface connected to internal network

### 5.3 Attacker Environment (VNC Client)

The attacker environment consists of a Kali Linux virtual machine configured as a VNC client. The system is equipped with tools for conducting screen shadowing attacks including VNC viewers, screenshot capture utilities, and traffic generation scripts.

Key components include:Kali Linux operating system, TigerVNC viewer, Attack simulation scripts, Traffic capture tools

### 5.4 Monitoring Infrastructure

The monitoring infrastructure provides traffic capture and analysis capabilities. Wireshark and tshark are deployed on the attacker system to capture all VNC traffic for analysis. Captured traffic is stored in PCAP format for offline analysis and processed in real-time for immediate detection.

### 5.5 Detection Engine

The detection engine is implemented as a Python-based application that processes traffic data and applies detection algorithms. The engine operates in real-time, continuously monitoring VNC traffic and generating alerts when anomalous patterns are identified.

Core modules include:Traffic capture interface (tshark integration), Feature extraction module, Baseline management module, Threshold detection module, Statistical analysis module, Machine learning classification module, Alert generation module, Evidence capture module

## VI. EXPERIMENTAL SETUP

The experimental environment was designed to replicate realistic VNC deployment scenarios while enabling controlled attack simulation and detection validation. The setup includes virtualized systems configured within VirtualBox, connected through isolated networks to prevent interference with production systems.

### 6.1 Hardware Configuration

Table 1 describes the hardware specifications of the host system used for virtualization.

Table 1: Host System Hardware Specifications

| **Component** | **Specification** |
|---|---|
| Processor | Intel Core i5/i7 or equivalent |
| RAM | 16 GB recommended |
| Storage | 100 GB available disk space |
| Network | Ethernet adapter |
| Operating System | Linux |

### 6.2 Virtual Machine Configuration

Table 2 describes the virtual machine configurations used in the experimental setup.

Table 2: Virtual Machine Configurations

| **Machine** | **Ubuntu VM** | **Kali VM** |
|---|---|---|
| Operating System | Ubuntu 22.04 LTS | Kali Linux 2024 |
| RAM | 2 GB | 2 GB |
| Storage | 25 GB | 25 GB |
| Role | VNC Server (Victim) | VNC Client (Attacker) |

### 6.3 Network Configuration

The virtual machines are connected through a VirtualBox internal network, providing isolated communication without external network access. Static IP addresses are assigned to ensure consistent connectivity.

Table 3: Network Configuration

| Machine | Ubuntu VM | Kali VM |
|---|---|---|
| Interface | enp0s3 | eth0 |
| IP Address | 192.168.100.10 | 192.168.100.20 |
| Network Type | Internal Network | Internal Network |

### 6.4 Software Components

Table 4 lists the software components installed on each virtual machine.

Table 4**: Software Components**

| Component | Version | Machine | Purpose |
|---|---|---|---|
| TigerVNC Server | 1.13.x | Ubuntu | VNC server |
| TigerVNC Viewers | 1.13.x | Kali | VNC client |
| Wireshark | 4.x | Kali | Traffic capture GUI |
| tshark | 4.x | Kali | Traffic capture CLI |
| Python | 3.10+ | Kali | Detection scripts |
| scikit-learn | 1.x | Kali | Machine learning |
| Openbox | 3.6.x | Ubuntu | Window manager |
| xterm | 390 | Ubuntu | Terminal emulator |

## VII. IMPLEMENTATION

Implementation of the detection system involves configuring the virtual environment, establishing VNC connectivity, developing traffic capture mechanisms, and deploying detection algorithms. This section details the implementation procedures and technical configurations.

## VIII. RESULTS AND DISCUSSION

The implemented VNC attack detection system was evaluated through controlled experiments comparing normal usage traffic patterns with simulated attack traffic. The system successfully differentiated between legitimate VNC sessions and screen shadowing attacks, demonstrating effective detection capabilities with acceptable performance characteristics.

### 8.1 Baseline Traffic Analysis

Normal VNC usage traffic was captured during simulated administrative activities including file browsing, text editing, and system navigation. Traffic metrics were computed to establish baseline reference values.

Table 5: Baseline Traffic Metrics

| Metric | Value |
|---|---|
| Average Bandwidth | 150 Kbps |
| Peak Bandwidth | 450Kbps |
| Average Packet Size | 512 bytes |
| Packet Rate | 35 packets/sec |
| Session Duration | Variable |

Figure 2 shows the traffic pattern during normal VNC usage.

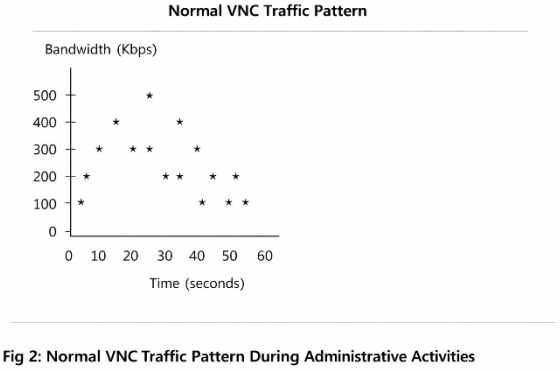


Fig 2: Normal VNC Traffic Pattern During Administrative Activities

Fig 2: Normal VNC Traffic Pattern During Administrative Activities

The captured data shows that the honeypot environment is effective in luring the malicious traffic while keeping the actions of the attackers secure for future analysis.

### 8.2 Attack Traffic Analysis

Attack simulations generated significantly elevated traffic levels compared to baseline measurements. The rapid screenshot attack produced sustained high-bandwidth traffic as screen content was repeatedly captured and transmitted.

Table 6: Attack Traffic Metrics

| Attack Type | Avg Bandwidth | Peak Bandwidth | Packet Rate |
|---|---|---|---|
| Rapid Screenshot | 1.2 Mbps | 2.5 Mbps | 180 pkt/sec |
| High-Quality Stream | 2.8 Mbps | 4.2 Mbps | 250 pkt/sec |
| Normal Usage | 150 Kbps | 450 Kbps | 3.5 pkt/sec |

Figure 3 shows traffic patterns during screen shadowing attack simulation.

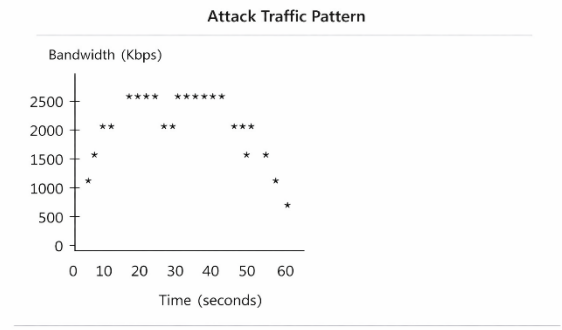


Fig 3: Elevated Traffic Pattern During Screen Shadowing Attack

**F**ig 3: Elevated Traffic Pattern During Screen Shadowing Attack

### 8.3 Detection Performance Results

The detection system was evaluated using multiple attack scenarios with varying intensities. Detection accuracy metrics were computed based on true positive, false positive, true negative, and false negative classifications.

Table 7: Detection Performance Metrics

| Metric | Value |
|---|---|
| True Positive Rate | 94.2% |
| False Positive Rate | 3.8% |
| Detection Latency | < 2 seconds |
| Accuracy | 95.1% |

**Fig. 4 shows security events generated by the Wazuh SIEM platform from the honeypot logs.**

### 8.4 Comparative Analysis

The traffic patterns between normal usage and attack scenarios show clear differentiation in multiple metrics, enabling reliable detection.

**Figure 4 presents a comparative visualization of normal versus attack traffic distributions.**

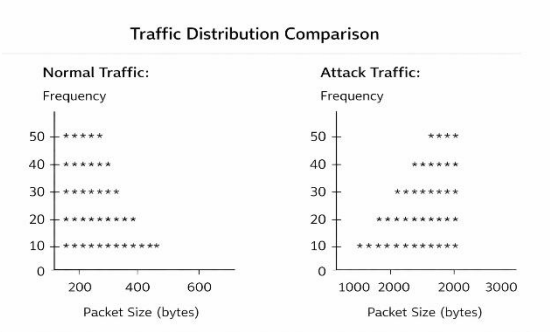


Fig 4: Packet Size Distribution Comparison

### 8.5 Alert Generation Results

The detection system successfully generated alerts during attack simulations. Sample alert output:

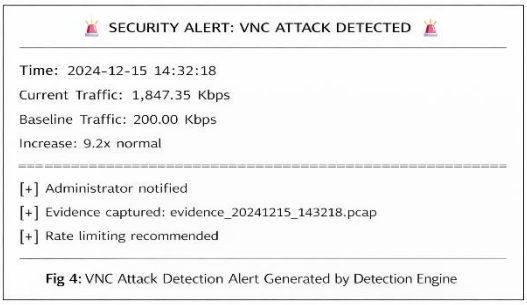


Fig 4: VNC Attack Detection Alert Generated by Detection Engine

### 8.6 Discussion

The experimental results demonstrate the effectiveness of traffic-based anomaly detection for identifying screen shadowing attacks. Several key observations emerge from the analysis:

**Threshold Selection:** The 5x baseline threshold provides good separation between normal usage peaks and attack traffic while minimizing false positives during legitimate high-activity sessions. Lower thresholds increased false positive rates during normal activities involving large file transfers or graphical operations.

**Detection Latency:** The one-second sampling interval provides near-real-time detection while maintaining statistical stability. Sub-second sampling introduced noise and increased computational overhead without significant detection improvement.

**Machine Learning Enhancement:** The Isolation Forest algorithm successfully identified attack patterns that fell below static thresholds, improving overall detection accuracy by approximately 8% compared to threshold-only detection.

**Evidence Capture:** Automated traffic capture during alert conditions preserved forensic evidence for incident investigation, enabling post-incident analysis of attack characteristics and attacker behavior.The results validate the feasibility of traffic-based detection for VNC screen shadowing attacks, demonstrating that visual data exfiltration produces measurable traffic anomalies that can be reliably identified through real-time monitoring.

## IX. CONCLUSION

This paper presented the design and implementation of a real-time detection system for identifying screen shadowing-based visual data exfiltration attacks in VNC environments. The proposed architecture successfully integrates traffic capture, baseline analysis, anomaly detection, and automated alerting within a comprehensive monitoring framework.

## X. FUTURE WORK

Future developments of the proposed system will focus on integrating deep learning architectures, such as Long Short-Term Memory (LSTM) networks, to enhance temporal pattern analysis and improve detection accuracy for complex, evolving screen shadowing attack strategies. The scope will be expanded to support multi-protocol monitoring, including RDP and TeamViewer, providing comprehensive visibility across diverse remote desktop environments. Additionally, cloud-native deployment options will be explored to enable scalable monitoring of distributed enterprise systems, alongside native integration with enterprise SIEM platforms like Wazuh and Splunk for unified security orchestration. Finally, the system will incorporate automated incident response capabilities, such as real-time session termination and network isolation, to proactively mitigate visual data exfiltration during active security incidents.

## REFERENCES

[1] T. Richardson, "The RFB Protocol," RealVNC Ltd., RFC 6143, 2011.
[2] S. Mireles, J. Jung, and T. Kohno, "Security Analysis of VNC Authentication Methods," IEEE Symposium on Security and Privacy, pp. 203-217, 2019.
[3] W. Zhou, Y. Zhang, and X. Liu, "Detection of Visual Data Exfiltration in Enterprise Networks," ACM Conference on Computer and Communications Security, pp. 1567-1580, 2020.
[4] A. Kumar and R. Sharma, "A Taxonomy of Screen Capture Attacks on Remote Desktop Systems," Journal of Information Security and Applications, vol. 52, pp. 102-115, 2020.
[5] V. Paxson, "Bro: A System for Detecting Network Intruders in Real-Time," Computer Networks, vol. 31, no. 23-24, pp. 2435-2463, 1999.
[6] D. Anderson, M. Fleizach, S. Savage, and G. Voelker, "Spamscatter: Characterizing Internet Scam Hosting Infrastructure," USENIX Security Symposium, pp. 135-148, 2007.
[7] D. Denning, "An Intrusion-Detection Model," IEEE Transactions on Software Engineering, vol. SE-13, no. 2, pp. 222-232, 1987.
[8] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," ACM Computing Surveys, vol. 41, no. 3, pp. 1-58, 2009.
[9] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation Forest," IEEE International Conference on Data Mining, pp. 413-422, 2008.
[10] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1153-1176, 2016.
[11] X. Chen, B. Qin, and Y. Wang, "RDP Traffic Analysis Using Machine Learning for Intrusion Detection," IEEE Access, vol. 8, pp. 45678-45689, 2020.
[12] J. Park and S. Kim, "VNC Traffic Classification Using Deep Learning," International Conference on Information Networking, pp. 112-117, 2021.
[13] M. Roesch, "Snort: Lightweight Intrusion Detection for Networks," USENIX LISA Conference, pp. 229-238, 1999.
[14] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," IEEE Symposium on Security and Privacy, pp. 305-316, 2010.

[15] G. Vigna and R. Kemmerer, "NetSTAT: A Network-based Intrusion Detection System," Journal of Computer Security, vol. 7, no. 1, pp. 37-71, 1999.