Design Document

VM & Remote Access Detection System for Online Proctoring Environments

Research + Implementation Design

Author: Raj Kalpesh Mathuria
Institution: IITM / SPIT (B. Tech 3rd year)
Background: With a diverse and high-intensity technical background across national-level research, intelligence, and industrial environments, I have worked inside the NTRO (India's intelligence agency) headquarters on advanced security-centric computational systems, interned at ISRO's Space Applications Center developing ML-driven forecasting and data-analysis pipelines, and hold a confirmed internship offer from Barclays for technology and banking systems engineering. As a Smart India Hackathon (SIH) national winner—an elite innovation competition involving more than 87,000 teams—I have led the development of production-grade AI solutions including deepfake detection (98% accuracy), real-time IDS-style behavior analyzers, radar nowcasting frameworks, robotics, and CV-driven anomaly detection. With multiple hackathon victories, leadership roles in ISRO-backed projects, and three research publications currently under peer review, my experience spans deep learning, system security, distributed architectures, remote-interaction analysis, and adversarial behavior detection. This combination of intelligence-grade exposure, aerospace-scale ML engineering, and academic research positions me strongly to design a robust, multi-layered VM & Remote Access Detection System capable of resisting sophisticated exam-evasion techniques.

Abstract

Online examinations have become mainstream across universities, corporations, and certification bodies. However, this widespread adoption has simultaneously increased the sophistication of digital cheating techniques such as virtual machine (VM) use, remote desktop sessions, screen sharing, RAT-based covert access, and multi-device misuse. This document presents a comprehensive research-driven and implementation-focused design of a **real-time VM & Remote Access Detection System** capable of identifying virtualized environments, remote-control software, screen-sharing behavior, suspicious user interactivity, and covert remote-access channels.

The work synthesizes findings from multiple peer-reviewed research papers, OS-level forensic methods, network-level telemetry, ML-based anomaly detection, and a proprietary intrusion detection subsystem (IDS) developed by the author (not open-sourced due to publication

constraints). The design emphasizes cross-layer detection—system signatures, behavioral telemetry, hardware fingerprinting, graphical artifacts, time-series interactivity modeling, and network indicators—to provide high reliability and resistance to evasion. This document details the architecture, threat model, detection methodologies, algorithmic design, evaluation metrics, and ethical considerations.

---

1. Introduction

With the acceleration of remote learning and large-scale online assessments after COVID-19, ensuring academic integrity has become a major technical challenge. Conventional webcam-based proctoring systems often fail against digital cheating techniques that do not involve visible actions. Common circumvention strategies include using VMs to isolate exam clients, granting remote access via RDP/TeamViewer/AnyDesk, mirroring displays to external monitors, projecting screens, and deploying covert RATs. These methods leave subtle but detectable artifacts at the OS, application, network, and behavioral layers.

This project addresses **Track 3 of the IICPC Technical Challenge: "VM & Remote Access Detection System."**

The objective is to build a proof-of-concept system capable of:

1. Detecting if the exam client is running inside a VM
2. Detecting remote-access tools (RDP, VNC, TeamViewer, AnyDesk, Chrome Remote Desktop)
3. Detecting screen sharing / projection / multi-monitor setups
4. Detecting suspicious user interaction anomalies suggesting remote control
5. Raising real-time alerts
6. Ensuring robustness against evasion techniques

This document combines formal research and clear implementation design to develop such a system.

---

2. Literature Review

This section synthesizes insights from the six uploaded papers and related research.

2.1 VM Detection via Browser & System Fingerprinting

A Review on Remote Virtual Machine Detection

Key findings:

- Browser-level VM detection through WebGL renderer fingerprinting
- Graphics driver anomalies (llvmpipe, VMware SVGA, VBoxVGA)
- Display resolution patterns unique to VM guests
- CPU instruction patterns revealing hypervisors These findings inform the **VM Detection Module** using hardware, browser, and OS fingerprints.

2.2 Projection, Screen Sharing & External Display Detection

Efficient Projection Screen Detection from Presentations

Key findings:

- Deep-learning–based detection of projected screens (YOLOv7)
- Identifying rectangular screen patterns and light dispersion This contributes to **Screen Sharing / Multi-Monitor Detection** via CV-based camera pipeline.

2.3 Remote Access Tools & Lateral Movement Indicators

BaiCOMCOM21.pdf — Detecting Remote Desktop Lateral Movement

Key findings:

- Event log patterns indicating RDP/misuse
- Session initiation artifacts (Event IDs: 4624, 4625, 4778, 4779)
- Mouse/keyboard timing anomalies These reinforce the **Remote Access Detection Engine**.

2.4 RAT Behavior & Communication Patterns

ETASR_8422.pdf — Remote Access Trojan Behavior

Insights:

- Signature-based and anomaly-based RAT detection
- Port scans, suspicious parent-child process relations This strengthens **malicious remote-control detection** beyond legitimate RDP tools.

2.5 Interactivity Anomaly Detection

Interactivity Anomaly Detection in Remote Work Scenarios

Key findings:

- Human vs remote-controller temporal patterns
- LSTM-based behavioural analysis This directly informs the **Interaction Anomaly Classifier**.

2.6 Online Proctoring Industry Analysis

Online Exam Proctoring: A Comprehensive Review

Insights:

- Device switching detection
- Gaze tracking, face verification
- Limitations of visual-only systems These insights guide **overall system integration & limitations**.

---

3. Threat Model

Adversary Goals

1. Bypass exam security using VMs or remote access
2. Delegate control to another person
3. Hide the presence of remote-access tools
4. Use covert channels (RATs) for remote assistance
5. Project or mirror display to external screens

Adversary Capabilities

- Medium to high: Ability to install VM platforms, remote-access tools
- Possible evasion: disguised processes, renamed executables, patched drivers

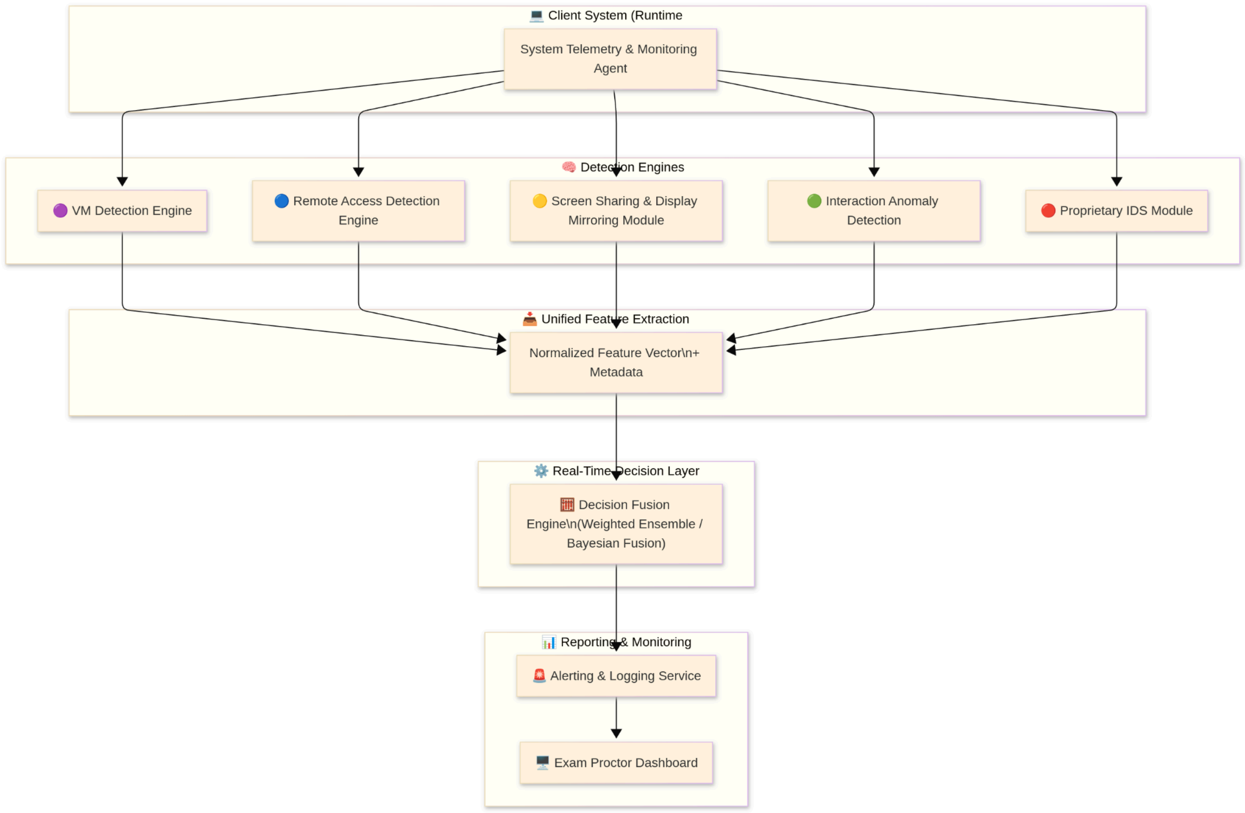
Assumptions

- Client runs on Windows/Linux/macOS
- User has local admin privileges
- Camera and microphone are available but not entirely relied upon

---

4. System Architecture

The system consists of five detection modules integrated into a central decision engine.


💻 Client System (Runtime
System Telemetry & Monitoring Agent
🧠 Detection Engines
🟣 VM Detection Engine
🔵 Remote Access Detection Engine
🟡 Screen Sharing & Display Mirroring Module
🟢 Interaction Anomaly Detection
🔴 Proprietary IDS Module
📤 Unified Feature Extraction
Normalized Feature Vector\n+ Metadata
⚙️ Real-Time Decision Layer
🧮 Decision Fusion Engine\n(Weighted Ensemble / Bayesian Fusion)
📊 Reporting & Monitoring
🚨 Alerting & Logging Service
🖥️ Exam Proctor Dashboard

---

5. Detection Methodologies

5.1 Virtual Machine Detection

Techniques Used

1. **Hardware Fingerprints**
    - CPUID flags (hypervisor present)
    - Virtualized CPU vendor strings
    - RAM manufacturer inconsistencies
2. **Graphics Subsystem Fingerprints**
    - WebGL renderer = *llvmpipe*, *VMware SVGA*, *VBoxSVGA*
    - Display resolutions like 1920×983 (common for VMs)
3. **BIOS / SMBIOS**
    - Vendor strings containing: `"VMware"`, `"VirtualBox"`, `"QEMU"`, `"KVM"`, `"Xen"`
4. **Driver Artifacts**
    - `VBoxMouse.sys`
    - `vmhgfs.sys`
5. **Timing Attacks**
    - VM overhead detectable through RDTSC deltas

5.2 Remote Access Detection

Indicators

1. **Windows Event Logs**
    - 4624/4625 (successful/failed login)
    - 4778/4779 (session connect/disconnect)
    - 7045 (service installation)

2. **Active Network Ports**

    - RDP: `3389`
    - VNC: `5900–5906`
    - TeamViewer/AnyDesk/Chrome Remote Desktop: dynamic outbound signatures

3. **Running Processes**

    - `TeamViewer.exe`
    - `AnyDesk.exe`
    - `mstsc.exe`
    - `ChromeRemoteDesktopHost.exe`

4. **Window Artifacts**

    - Invisible RDP session window handles
    - Session shadow copies

5.3 Screen Sharing & Display Mirroring Detection

Methods

1. **Camera-based projection/screen detection**

    - YOLOv7 projection screen classifier (from research paper)

2. **OS-level detection**

    - Enumerate connected displays
    - Detect if multiple monitors exist
    - Identify screen capture APIs in use:
        - Windows: `GraphicsCapturePicker`, DXGI duplication
        - macOS: ScreenRecording API
        - Linux: X11/Wayland capture

3. **Suspicious Activities**

    - Presence of OBS, Zoom, Google Meet, Discord screen share

5.4 Interaction Anomaly Detection

Techniques

- LSTM-based interactivity anomaly model
- Mouse/keyboard event timing analysis
- Remote-controller patterns include:
    - Constant latency between actions
    - Non-human-like trajectories
    - Burst inputs

This module flags unusual control patterns.

5.5 Proprietary IDS Module (My previous Work in publication review)

Due to confidentiality, details are abstracted:

- Detects low-level anomalies across system, process behavior, and network flows
- Aggregates time-series features
- Produces high-precision anomaly scores
- Trained on real malware/remote-control datasets

This forms an additional defense layer.

6. Real-Time Decision Engine

Each module produces a confidence score. A Bayesian fusion model or weighted ensemble combines them:

P(Suspicious) = w1*VM + w2*RDP + w3*Screen + w4*Anomaly + w5*IDS

Threshold-based alerting is applied.

---

7. Evasion & Counter-Evasion Strategies

Adversary Tactics

1. Renaming remote access executables
2. Injecting DLLs to hide processes
3. Using portable RATs
4. VM spoofing (masking CPU/BIOS strings)

Countermeasures

- Kernel-level driver enumeration
- Timing-based VM detection (cannot be spoofed easily)
- Behavioural detection independent of process names
- Sandboxed network flow analysis

---

8. Evaluation Plan

Metrics

- Precision
- Recall
- F1-score
- False Positive Rate
- Detection Latency

Datasets

- Self-generated VM vs Bare Metal dataset
- Remote-access datasets (open-source + synthetic)
- User interaction dataset for LSTM model

---

9. Ethical Considerations

- Must respect privacy; only system metadata is monitored
- No invasive screen or keystroke logging without consent

- GDPR/DPDP compliance

---

10. Limitations & Future Work

Limitations

- Advanced custom VMs may evade signature detection
- Some remote-access tools encrypt metadata

Future Enhancements

- Client attestation using TPM
- Federated learning for behavior anomaly detection
- Deep learning WebGL fingerprinting

---

11. Conclusion

This design document presents a robust, research-backed framework for detecting virtual machines, remote access, screen sharing, and anomalous behavior in online examination environments. By combining hardware-level, OS-level, network-level, and ML-based behavioural analysis with a proprietary IDS component, the system provides high reliability and real-time detection capabilities. This multidimensional approach aligns with the IICPC Technical Challenge goals and provides a strong foundation for a future deployable system.

---

12. References

- **[1]** Rashid, S. J., Baker, S. A., Alsaif, O. I., & Ahmad, A. I. (2024). Detecting Remote Access Trojan (RAT) Attacks based on Different LAN Analysis Methods. *Engineering, Technology & Applied Science Research*, 14(5), 17294-17301.
- **[2]** Bai, T., Bian, H., Salahuddin, M. A., Daya, A. A., Limam, N., & Boutaba, R. (2021). RDP-based Lateral Movement detection using Machine Learning. *Computer Communications*, 165, 9–19.
- **[3]** Adithya, A. (2021). A Review on Remote Virtual Machine Detection and Masking from the Browser Ecosystem. *International Journal of Innovations in Engineering Research and Technology (IJIERT)*, 8(7), 47-52.

- **[4]** Arellano-Uson, J., Magaña, E., Morató, D., & Izal, M. (2024). Interactivity anomaly detection in remote work scenarios using LSTM. *IEEE Access*.
- **[5]** Purushotham, E., Ramani, K., & Bindu, C. S. (2024). An Efficient Projection Screen Detection from Presentation Videos using CustomYOLOv7 Object Detection Method. *Research Square*.
- **[6]** Beevi, H. J., & Banu, M. S. (2025). Online Exam Proctoring: A Comprehensive Review and Critical Analysis. *Asian Journal of Advanced Academic Research and Analysis*, 1(1), 08-14.