TELEGRAM-BASED ENDPOINT MONITORING AND REMOTE ADMINISTRATION FR

Institutional Research Paper Format
Student: [Your Name] Register No.: [Your Number]
Department: [Dept] Institution: [Institution]
Guide: [Guide Name] Academic Year: [YYYY-YYYY]

Abstract

This paper presents a Telegram-based endpoint monitoring framework designed for controlled cybersecurity research. The system supports telemetry collection, remote command execution, multi-device routing, and defensive analysis. Key modules include keystroke and clipboard capture, system profiling, screenshot/webcam acquisition, shell control, Wi-Fi profile enumeration, and file retrieval. Evaluation in a lab environment indicates responsive text-based command handling and predictable media-transfer delay. The project is documented with strict ethical guidance and blue-team mitigation recommendations.

1. Introduction

Trusted messaging APIs can be abused as command channels because traffic appears legitimate. This work builds a realistic final-year prototype to study both offensive workflows and defensive detection points. Objectives include modular design, multi-endpoint control, performance evaluation, and institutional compliance.

2. Literature Context

Prior studies and ATT&CK mappings show recurring phases: deployment, persistence, collection, execution, and exfiltration. Telegram APIs reduce setup complexity and support command plus file response channels, making them suitable for lab-scale emulation.

3. Proposed System Architecture

The framework follows an Operator -> Telegram API -> Endpoint Agent model. The agent contains modular collectors and handlers. Figure 1 provides the architecture diagram in renderable diagram format.

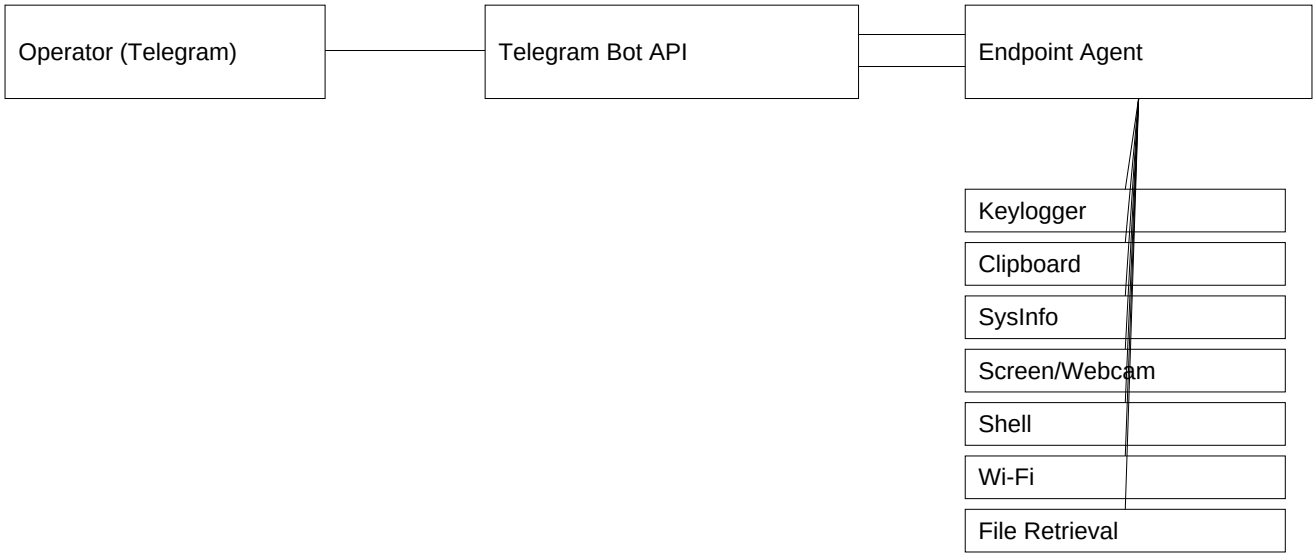

Figure 1: High-level architecture and module mapping.

4. Implementation and Command Workflow

Implementation uses Python modules for runtime control, command parsing, collection tasks, and response upload. Device IDs allow one operator chat to manage multiple endpoints safely in a lab.

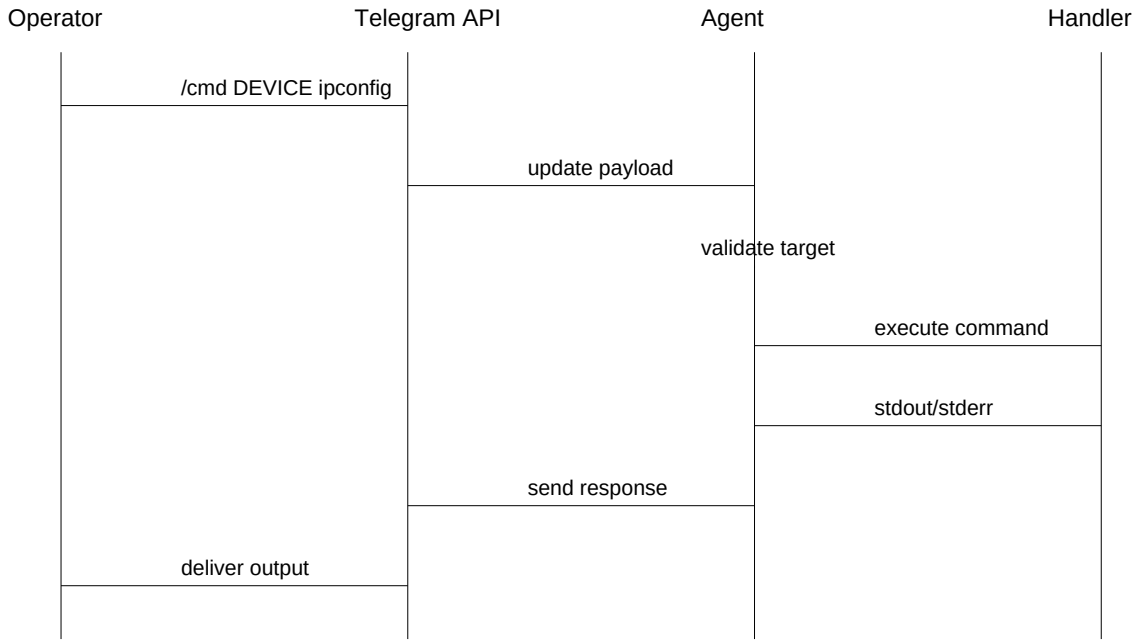

Figure 2: Command-processing sequence.

5. Evaluation

Evaluation criteria: command latency, success rate, media transfer practicality, and host resource footprint. Text commands generally return quickly; media commands take longer due to capture and upload overhead. Observed limits include network outages, webcam permission failures, and payload size constraints.

6. Security Analysis and Defensive Insights

The prototype maps to common ATT&CK behaviors and is valuable for defender training. Recommended controls include startup-change alerts, behavioral EDR rules, process lineage monitoring, and egress policy restrictions for unauthorized Telegram traffic.

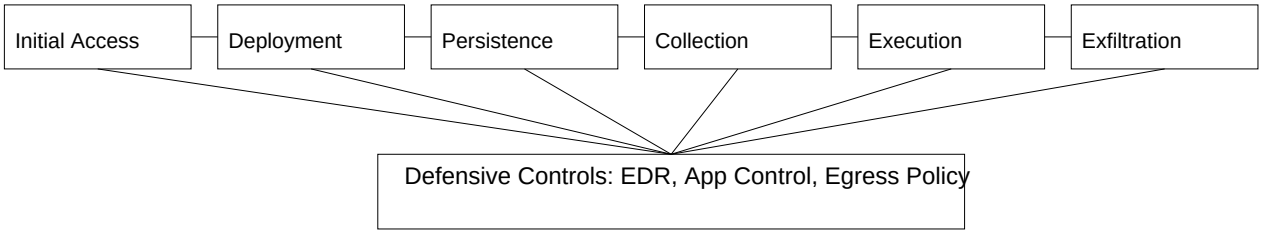

Figure 3: Risk lifecycle with defensive interception points.

7. Ethical and Institutional Compliance

This research must be executed only in authorized environments with written consent, faculty supervision, and non-sensitive data. All artifacts should be securely archived or deleted after assessment.

8. Conclusion and Future Work

The project demonstrates a complete, multi-module endpoint monitoring framework suitable for cybersecurity education and adversary emulation. Future work: stronger key management, cross-platform agent support, safer opt-in execution, and automated blue-team analytics dashboards.

References

[1] MITRE ATT&CK Enterprise Matrix.
[2] Telegram Bot API Documentation.
[3] NIST SP 800-53.
[4] Microsoft Windows Security Baselines.
[5] OWASP Logging Cheat Sheet.