IJCE; INTERNATIONAL JOURNAL OF CYBER ENGINEERING

E-ISSN: XXXX-XXXX
Vol. 01, No. 01, December, 2025, PP: 13-27
DOI:

ARTICLE

Network Forensic Analysis of a Multi-Stage Async RAT Infection Chain

Febrian Tri Hardiyanto *

Department of Informatics, Universitas Siber Muhammadiyah, Yogyakarta, Indonesia

***Corresponding author:** febrian20230100072@sibermu.ac.id

Abstract

This study investigates the complete infection chain of a multi-stage malware attack culminating in the deployment of the Async Remote Access Trojan (RAT). The primary objective is to identify critical stages of the infection process and derive Indicators of Compromise (IoCs) that can inform effective detection and prevention strategies. Contemporary malware increasingly employs sophisticated delivery mechanisms to evade conventional security defenses, presenting significant challenges for analysts due to multi-stage execution, obfuscation, and covert communication channels. A controlled virtual environment was established to safely execute and monitor the malware sample. Utilizing the Wireshark protocol analyzer and host-based artifact collection, the infection was traced from an ISO disk image file to a Windows Script File (WSF) loader. The analysis revealed a two-stage network communication process: an initial connection to a staging server for payload retrieval, followed by the establishment of an encrypted TLSv1.0 channel to a dynamic DNS-based Command and Control (C2) server operating on a non-standard port. The findings provide distinct IoCs, including network signatures and host-based traces, that can be leveraged for proactive defense. This research demonstrates that Async RAT infections rely on layered delivery mechanisms and encrypted communication channels to bypass detection. The identified IoCs offer actionable intelligence for security practitioners, contributing to the development of robust detection and mitigation strategies against evolving malware threats.

Keywords: Async RAT, Malware infection chain, Indicators of Compromise (IoCs), Command and Control (C2) communication, Network traffic analysis.

1. Introduction

Remote Access Trojans (RATs) represent a critical category of malicious software that provides adversaries with persistent, interactive access to compromised systems, enabling a wide range of malicious activities including data exfiltration, system surveillance, and remote command execution (Aburbeian et al., 2025). The sophistication and prevalence of RAT attacks have escalated substantially in recent years, with AsyncRAT emerging as one of the most widely deployed and technically advanced variants in the contemporary threat landscape. This malware family demonstrates sophisticated anti-analysis capabilities, including reflective code loading, comprehensive sandbox evasion techniques, and registry-based persistence mechanisms that significantly complicate detection efforts (Guo et al., 2020).

The operational impact of RAT campaigns is substantial and often underestimated. Longitudinal analysis of RAT ecosystems has revealed the extensive scope of these threats, with individual campaigns compromising tens of thousands of victims over extended periods while maintaining operational stealth (Wilkens et al., 2021). AsyncRAT exemplifies the evolutionary trajectory of modern malware development, incorporating modular architecture, encrypted command-and-control (C2) communications, and multi-stage infection chains that distribute attack activities across temporally dispersed phases to evade detection systems optimized for identifying concentrated bursts of suspicious activity.

Contemporary RAT infections typically manifest as multi-stage attack chains comprising distinct operational phases aligned with established cyber kill chain frameworks. These phases—initial access, execution, persistence establishment, C2 communication, lateral movement, and data exfiltration—may unfold over extended temporal periods and span multiple compromised hosts within target environments (Hossain et al., 2020). This temporal dispersion introduces significant challenges for detection systems, as individual stages may generate weak or ambiguous signals that appear benign when examined in isolation but constitute clear indicators of compromise when analyzed within the broader attack sequence context.

Network forensic analysis is essential for detecting and investigating RAT infections, as it leverages the unavoidable need for network communication in these attacks (Krych & Acosta, 2020). Studies show that early byte-level traffic inspection can achieve detection accuracies above 95% (Pi et al., 2023), while hybrid frameworks integrating host telemetry and network features report True Positive Rates over 93% with False Positive Rates below 0.2% (Guo et al., 2020), highlighting the effectiveness of integrated approaches (Wilkens et al., 2021).

The challenge of detecting encrypted RAT communications has motivated development of representation learning approaches that operate on traffic metadata and structural features rather than payload content. Recent ensemble methods combining flow-sequence analysis, payload embeddings, and recurrent neural networks have achieved precision and recall rates approaching 97% on encrypted RAT traffic benchmarks, demonstrating efficacy in scenarios where traditional deep packet inspection proves ineffective (Zhang et al., 2023). Additionally, automated signature generation frameworks have been developed to extract recurring byte sequences from RAT traffic and generate intrusion detection system rules, substantially reducing the manual effort required to maintain detection capabilities against evolving malware variants (Li et al., 2024).

Despite these advances, AsyncRAT detection and forensic analysis confront several significant technical challenges. The malware employs comprehensive anti-analysis techniques including reflective code loading, sandbox detection, and virtual machine awareness that collectively reduce the fidelity of dynamic analysis artifacts (BehradFar et al., 2020). The encryption of C2 communications and strategic abuse of legitimate cloud services for command-and-control infrastructure substantially reduce the effectiveness of signature-based detection approaches. Furthermore, incomplete telemetry resulting from selective logging or deliberate adversarial manipulation limits the completeness of forensic reconstructions, as the ability to reconstruct complete attack timelines depends critically on comprehensive, synchronized collection of both network and host-level artifacts (Zeng et al., 2022).

Instrumentation-free forensic systems have demonstrated the feasibility of comprehensive behavioral reconstruction with minimal performance overhead. Recent frameworks leveraging Event Tracing for Windows (ETW) have achieved approximately 90% True Positive Rates in cross-family RAT detection evaluations while imposing only 3.7% runtime overhead, demonstrating that comprehensive forensic capabilities need not impose prohibitive performance penalties (Zeng et al., 2022). The integration of network forensic methodologies into operational security workflows through structured frameworks has proven valuable for incident response, providing systematic approaches to evidence collection, timeline reconstruction, and attack source identification (Kusuma et al., 2021).

This study conducts a network forensic analysis of multi-stage AsyncRAT infections, focusing on behavioral characterization, command-and-control (C2) protocol examination, and the extraction of actionable indicators of compromise (IoCs). Using controlled malware execution, traffic capture, feature extraction, and timeline reconstruction, the research aims to document infection progression, analyze C2 mechanisms, optimize early detection features, and generate IoCs—such as signatures and behavioral patterns—for integration into security monitoring systems.

This research contributes to the broader body of knowledge in malware forensics and network security by providing detailed empirical analysis of AsyncRAT infection chains, demonstrating practical forensic methodologies applicable to contemporary RAT threats, and generating detection artifacts that enhance defensive capabilities against this significant threat category. The findings are expected to inform both academic research directions and operational security practices, advancing the state of the art in RAT detection and forensic analysis.

2. Literature Review

2.1 Remote Access Trojan Evolution and Characteristics

Remote Access Trojans have undergone substantial evolutionary development over the past decade, transitioning from rudimentary backdoor mechanisms into sophisticated, modular frameworks that employ advanced evasion techniques and persistence mechanisms. Contemporary RAT families, including AsyncRAT, demonstrate technical capabilities that substantially complicate detection and forensic analysis efforts. Guo et al. (2020) documented that modern RATs employ reflective code loading mechanisms, comprehensive sandbox detection capabilities, and registry-based persistence techniques to maintain long-term command-and-control access while evading conventional security monitoring systems. These anti-analysis techniques include virtual machine awareness, debugger detection, and behavioral adaptations designed to thwart dynamic analysis in controlled environments.

The operational characteristics of RAT campaigns reveal substantial scope and persistence. Rao et al. (2024) conducted MITRE-aligned analysis of prominent malware instances, documenting the sophisticated tactics, techniques, and procedures (TTPs) employed by contemporary RAT operators. Their analysis revealed that modern RATs establish modular C2 channels that support incremental capability deployment, enabling adversaries to adapt their post-exploitation activities based on victim environment characteristics and operational objectives. The persistence mechanisms employed by AsyncRAT variants include registry autorun modifications, scheduled task creation, and service installation, collectively ensuring that malware survives system reboots and maintains operational continuity across extended campaign durations.

Large-scale ecosystem analyses have provided empirical evidence of RAT operational scope that would remain invisible through conventional endpoint monitoring. Wilkens et al. (2021) examined RAT controller databases and victim telemetry, revealing campaigns that compromised tens of thousands of systems while employing hostname churn techniques and distributed infrastructure to hinder traceability and remediation efforts. These findings underscore the necessity for comprehensive network forensic capabilities that can reconstruct attack chains and attribute activities to specific threat actor infrastructure despite deliberate obfuscation attempts.

2.2 Machine Learning and Deep Learning Detection Approaches

The application of machine learning and deep learning techniques to RAT detection has proliferated substantially in recent literature, with researchers exploring diverse feature engineering strategies, model architectures, and hybrid approaches that combine multiple data sources. Guo et al. (2020) proposed the Phased Remote Access Trojan Detection (PRATD) methodology, which recognizes that RAT behavior evolves across distinct operational phases and implements phase-specific detection models trained on both host-based and network-based features. Evaluated across 20 RAT families and benign applications, PRATD achieved True Positive Rates of 93.609% for known RAT variants with a False Positive Rate of 0.407%, while maintaining 81.928% TPR for previously unseen RAT families with FPR of 0.185%. This phased approach demonstrates that stage-aware detection models can substantially improve classification performance compared to monolithic detection frameworks.

Deep learning architectures have demonstrated particular efficacy for early-stage RAT detection using minimal network traffic context. Pi et al. (2023) developed a novel methodology that encodes early TCP payload byte sequences into 256×256 byte transition probability matrices—fixed-size Markov representations that capture byte-to-byte transition patterns independent of payload semantics. These matrices are subsequently processed by Convolutional Neural Networks trained to distinguish malicious from benign traffic patterns. Evaluated on 61 RAT families and 58 benign applications, this approach achieved 95.5% detection accuracy using only the first few server packets of a connection, substantially reducing time-to-detection compared to traditional methods requiring

extended traffic captures. The byte-level representation maintains robustness to payload encryption and obfuscation by focusing on structural patterns rather than semantic content.

Hybrid feature engineering approaches that combine static and dynamic analysis artifacts with optimized feature selection have achieved exceptional performance in controlled experimental settings. Aburbeian et al. (2025) developed a comprehensive RAT detection framework employing information gain and correlation-based feature selection algorithms to identify optimal feature subsets from extensive static and dynamic analysis. Their approach, evaluated using ensemble machine learning and deep learning models, achieved accuracy of 99.75% with a false alarm rate of 0.3% on sandbox-derived datasets. However, the authors acknowledged that generalization to operational environments characterized by diverse benign application behaviors and sophisticated adversarial evasion remains an ongoing challenge requiring additional validation.

2.3 Encrypted Traffic Analysis and Representation Learning

The increasing prevalence of encrypted command-and-control communications has motivated development of detection approaches that operate on traffic metadata and structural features rather than payload content inspection. Zhang et al. (2023) introduced ER-ERT (Ensemble Representation learning of Encrypted RAT Traffic), which addresses the challenge of detecting RAT communications in encrypted network flows through multi-perspective representation fusion. The framework combines three complementary components: (1) flow-sequence temporal embeddings generated via CNNs and Reproducing Kernel Hilbert Space (RKHS) transformations, (2) payload spatial embeddings learned through autoencoders and bidirectional Gated Recurrent Units, and (3) stage-based behavioral attributes that capture attack progression patterns. Evaluated on encrypted RAT traffic benchmarks, ER-ERT achieved precision of 97.0% and recall of 96.5%, demonstrating substantial improvement over single-representation baselines and validating the efficacy of ensemble approaches for encrypted traffic analysis.

The representation learning paradigm enables detection systems to learn discriminative features directly from traffic observations rather than relying on manually engineered signatures that may be brittle against polymorphic techniques. This approach proves particularly valuable for encrypted communications where traditional deep packet inspection cannot access payload semantics. However, representation learning methods require substantial labeled training datasets to achieve robust generalization, and their performance may degrade when confronted with RAT variants employing novel communication patterns not represented in training data (Zhang et al., 2023).

2.4 Network Traffic Analysis Methodologies

Network traffic analysis for Remote Access Trojan (RAT) detection can be approached at different levels, including byte-level payload inspection, flow-level statistical characterization, and application-level behavioral profiling. Research has shown that examining initial TCP packets during connection establishment enables rapid and accurate RAT detection by leveraging protocol-specific patterns present before significant data transfer occurs (Pi et al., 2023). In addition, an automated framework known as Parallel-SWSA (Sliding Window Sequence Alignment) has been introduced to segment traffic, apply sequence alignment for recurring byte patterns, and translate them into Snort-compatible rules (Li et al., 2024). Validated across diverse datasets, this approach demonstrated high detection accuracy, resilience to noise, and a substantial reduction in manual effort required to maintain signature databases against evolving malware variants.

Application-level behavioral profiling provides complementary detection capabilities by modeling expected network behavior for legitimate applications and flagging deviations indicative of compromise or malicious injection. Krych and Acosta (2020) demonstrated practical approaches for uncovering and decoding malware communications using scriptable network analysis tools, emphasizing the value of protocol-aware inspection that understands application-layer semantics. Their work illustrated that combining automated analysis with expert-guided investigation enables efficient identification of anomalous communications embedded within legitimate protocol traffic.

2.5 Multi-Stage Attack Detection and Forensic Reconstruction

The detection of multi-stage attack campaigns requires analytical frameworks capable of correlating temporally dispersed events across multiple hosts and reconstructing coherent attack narratives from low-level security alerts. Wilkens et al. (2021) introduced Kill Chain State Machines (KCSM), which model attack progression through cyber kill chain phases and aggregate individual security alerts into stage-labeled scenario graphs. KCSM employs directional analysis of alert relationships to infer attack stage transitions and synthesizes high-level scenario representations that enable analysts to comprehend complex, multi-phase campaigns while reducing alert

volumes by up to three orders of magnitude. This abstraction from individual alerts to coherent attack scenarios represents a critical advancement in managing the cognitive burden imposed on security operations personnel by high-volume alert streams characteristic of enterprise environments.

Forensic reconstruction of RAT behaviors requires comprehensive capture of both network communications and host-level system activities to enable accurate timeline construction and behavioral attribution. Zeng et al. (2022) developed RATScope, an instrumentation-free Windows forensic framework that leverages Event Tracing for Windows (ETW) to record semantic RAT behaviors without invasive instrumentation that might be detected by anti-analysis mechanisms. RATScope implements program behavior modeling techniques that reconstruct high-level RAT semantics—including file operations, registry modifications, and network communications—from low-level system events. Evaluated across diverse RAT families spanning multiple years, RATScope achieved approximately 90% True Positive Rate in cross-family detection experiments and maintained 80% TPR in temporal validation spanning two years, while imposing only 3.7% runtime overhead. These performance characteristics demonstrate that comprehensive forensic capabilities can be achieved without prohibitive performance penalties, making such approaches viable for deployment in production environments.

The integration of causal analysis with semantic modeling provides additional capabilities for reconstructing attack narratives from incomplete or noisy telemetry. Hossain et al. (2020) developed SLEUTH, a system for real-time attack scenario reconstruction from commercial off-the-shelf (COTS) audit data. SLEUTH mines causal relationships among system events, constructs provenance graphs representing information flow dependencies, and applies statistical anomaly detection to identify suspicious causal chains indicative of multi-stage attacks. This approach enables reconstruction of attack scenarios even when individual events appear benign in isolation, addressing the fundamental challenge that multi-stage attacks deliberately distribute suspicious activities across temporal and spatial dimensions to evade detection.

2.6 Datasets, Evaluation Methodologies, and Performance Benchmarks

The datasets and evaluation methodologies employed in RAT detection research exhibit substantial heterogeneity, complicating direct performance comparisons across studies. Pi et al. (2023) employed a curated dataset comprising 61 RAT families and 58 benign applications, focusing evaluation specifically on early-stage traffic to assess detection timeliness. Guo et al. (2020) constructed datasets combining host telemetry and network traffic across 20 RAT families and benign programs, enabling evaluation of hybrid detection approaches. The diversity of dataset construction methodologies—including malware sample sources, traffic capture strategies, and benign baseline selection—introduces variability that affects reported performance metrics and generalization characteristics.

Evaluation practices in contemporary RAT detection research emphasize multiple performance dimensions including accuracy, precision, recall (True Positive Rate), False Positive Rate, and in some cases computational overhead. Zeng et al. (2022) notably included runtime overhead measurements (3.7%) alongside detection performance metrics, recognizing that operational viability depends on both detection efficacy and system performance impact. However, many studies focus primarily on classification accuracy without comprehensive evaluation of computational requirements, deployment constraints, or performance degradation under adversarial conditions. This gap between controlled experimental evaluation and operational deployment requirements represents an ongoing challenge in translating research advances into production security systems.

Cross-validation practices vary substantially across studies, with some employing temporal validation (training on earlier samples and testing on more recent variants) to assess robustness against evolving threats, while others use random partitioning that may overestimate generalization performance. Guo et al. (2020) explicitly evaluated performance on both known and unknown RAT families to assess generalization capabilities, revealing substantial performance differences (93.6% TPR for known vs. 81.9% TPR for unknown families) that highlight the challenge of detecting novel variants. These methodological choices substantially affect reported performance characteristics and the validity of conclusions regarding operational effectiveness.

2.7 Current Limitations and Research Gaps

Despite substantial advances in RAT detection and forensic analysis capabilities, several fundamental limitations persist in current methodologies. First, temporal drift and adversarial adaptation degrade detection performance over time as attackers evolve their techniques in response to deployed defenses. Zeng et al. (2022) documented performance degradation in temporal validation experiments, with cross-family detection TPR declining from 90% to 80% over a two-year span, illustrating the challenge of maintaining detection efficacy against evolving threats without continuous model retraining.

Second, the scarcity of standardized, publicly available datasets for multi-stage RAT attacks impedes reproducible research and independent validation of proposed methodologies. Many studies rely on privately collected datasets or synthesized attack scenarios that may not capture the full complexity and diversity of operational RAT campaigns (Aburbeian et al., 2025; Pi et al., 2023). The absence of benchmark datasets with ground truth labels for multi-stage attack progression limits comparative evaluation and hinders identification of methodological best practices.

Third, encrypted command-and-control communications and strategic abuse of legitimate cloud services for C2 infrastructure continue to challenge signature-based and content-inspection approaches. While representation learning methods demonstrate improved performance on encrypted traffic (Zhang et al., 2023), they require substantial labeled training data and may struggle to generalize to novel encryption schemes or protocol encapsulation techniques. The increasing prevalence of encryption across both legitimate and malicious traffic necessitates continued development of metadata-based and behavioral detection approaches.

Finally, computational overhead and deployment constraints receive insufficient attention in much of the literature. Zeng et al. (2022) represent a notable exception in explicitly measuring runtime overhead, but broader evaluation of model inference costs, memory footprints, and scalability characteristics remains sparse. Operational deployment requires detection systems that balance accuracy with computational efficiency, particularly in resource-constrained environments or high-throughput network monitoring scenarios. Future research must address these practical constraints to facilitate translation of academic advances into operational security capabilities.

3. Research methods

3.1 Research Design and Overview

This research employs a systematic experimental approach to investigate multi-stage AsyncRAT infection chains through comprehensive network forensic analysis. The methodology integrates controlled malware execution, multi-dimensional traffic capture, feature extraction, and forensic reconstruction techniques following established practices in malware analysis and network security research (Guo et al., 2020; Zeng et al., 2022). The research design encompasses five principal phases: (1) malware sample acquisition and validation, (2) controlled experimental environment deployment, (3) comprehensive network traffic capture and preservation, (4) multi-dimensional feature extraction and analysis, and (5) forensic timeline reconstruction and validation. This systematic approach enables rigorous characterization of AsyncRAT behavioral patterns across all infection lifecycle stages while maintaining experimental reproducibility and methodological transparency.

3.2 Malware Sample Collection and Preparation

3.2.1 Sample Acquisition Protocol

Malware samples were obtained from established repositories following protocols documented in contemporary RAT research (Aburbeian et al., 2025; BehradFar et al., 2020). Primary sources included VirusTotal and VirusShare, which provide verified malware specimens with comprehensive metadata including cryptographic hash values, antivirus detection ratios, and behavioral indicators. Sample selection criteria prioritized AsyncRAT variants demonstrating multi-stage infection characteristics, confirmed through preliminary static analysis and signature matching against known AsyncRAT indicators of compromise. Temporal diversity was incorporated by acquiring variants spanning multiple months to capture evolutionary changes in AsyncRAT implementations and evasion techniques.

Each acquired sample underwent verification procedures to ensure authenticity and family membership. Cryptographic hashes (MD5, SHA-1, SHA-256) were computed and cross-referenced against threat intelligence databases to confirm provenance and variant classification. Static analysis using PE (Portable Executable) inspection tools including PEiD, Detect It Easy, and custom YARA rules specifically crafted for AsyncRAT identification was conducted to validate family membership prior to dynamic analysis (BehradFar et al., 2020). Samples failing verification criteria or exhibiting ambiguous classification were excluded from the experimental corpus to maintain dataset integrity.

3.2.2 Benign Baseline Construction

A control dataset comprising benign applications was curated to establish behavioral baselines and enable comparative analysis. Benign samples were selected to represent diverse legitimate software categories including productivity applications, system utilities, and network-enabled services that generate traffic patterns potentially

similar to malicious communications. This diversity ensures that detection methodologies developed through the research can discriminate AsyncRAT traffic from legitimate application behaviors rather than simply identifying any network activity as suspicious (Guo et al., 2020; Pi et al., 2023).

3.3 Experimental Infrastructure and Controlled Environment

3.3.1 Isolated Laboratory Network Architecture

The experimental infrastructure comprised an isolated virtual network environment designed to safely execute malware samples while capturing comprehensive telemetry without risk of external propagation. The architecture implemented network segmentation using VLANs to separate infected hosts, simulated command-and-control servers, and monitoring infrastructure, following security best practices for malware analysis laboratories (Kusuma et al., 2021; Udeshi et al., 2025). Physical network isolation was enforced through dedicated hardware with no external connectivity, eliminating risks of inadvertent malware escape or communication with operational threat actor infrastructure.

The network topology incorporated multiple segments: (1) victim host systems executing various Windows operating system versions to assess cross-version compatibility and behavioral variations, (2) simulated internet gateway and DNS servers configured to facilitate malware execution without external network access, (3) network monitoring infrastructure positioned at strategic network segments to capture traffic from multiple vantage points, and (4) centralized log aggregation servers for correlation of multi-source telemetry. All systems were deployed as virtual machines using VMware ESXi and VirtualBox hypervisors, enabling rapid snapshot-based restoration between experimental trials to ensure consistent initial conditions (Udeshi et al., 2025).

3.3.2 Sandbox Environment Configuration

Malware execution occurred within carefully configured sandbox environments incorporating network emulation and comprehensive telemetry capture capabilities. The sandbox design followed principles established in contemporary malware analysis research, utilizing FakeNet-style network emulation to simulate internet services and command-and-control infrastructure responses (Udeshi et al., 2025). This approach enabled malware to execute normally and progress through infection stages while preventing actual external communications—a critical requirement for safely analyzing network-dependent malware behaviors in controlled environments.

Host-level instrumentation leveraged Event Tracing for Windows (ETW) to capture semantic system behaviors with minimal overhead, following methodologies validated in forensic RAT analysis research (Zeng et al., 2022). ETW providers were configured to log process creation events, file system modifications, registry operations, and network socket activities, providing comprehensive host-level context for correlation with captured network traffic. The instrumentation approach prioritized non-invasive monitoring to avoid triggering anti-analysis mechanisms commonly employed by sophisticated RAT variants. Zeng et al. (2022) demonstrated that ETW-based instrumentation imposes only 3.7% runtime overhead while maintaining high-fidelity behavioral capture, validating this approach for production-grade forensic analysis.

3.4 Network Traffic Capture and Data Collection

3.4.1 Multi-Point Packet Capture Strategy

Network traffic capture employed a multi-point strategy to ensure comprehensive visibility across all stages of the infection chain. Capture points were strategically positioned at the victim host network interface, the network gateway, and the simulated C2 server interface, enabling triangulation of network behaviors and detection of any evasion techniques (Krych & Acosta, 2020; Kusuma et al., 2021). This redundant capture approach guards against packet loss and provides multiple perspectives on network communications, facilitating validation of captured data completeness.

Packet capture utilized tcpdump and Wireshark to record full packet payloads in PCAP format, preserving complete protocol headers and payload content for subsequent analysis. Capture filters were initially configured broadly to record all traffic, with post-capture filtering applied during analysis phases to focus on specific protocols or communication patterns of interest. Each experimental trial generated timestamped PCAP files with associated metadata documenting the sample cryptographic hash, execution timestamp, experimental conditions, and observed behavioral characteristics (Kusuma et al., 2021).

3.4.2 Temporal Resolution and Early-Stage Capture

Recognizing that multi-stage infections unfold over extended time periods, capture sessions were maintained for sufficient duration to observe complete infection lifecycles, including initial compromise, C2

establishment, payload staging, and post-exploitation activities. Preliminary experiments established baseline capture durations, which were subsequently adjusted based on observed malware dormancy periods and staged payload delivery timings. Fine-grained temporal analysis required capturing early-stage network behaviors with microsecond-precision timestamps to enable detailed temporal correlation and sequence analysis (Pi et al., 2023).

Particular emphasis was placed on capturing initial TCP handshakes and first application-layer exchanges, as recent research has demonstrated that minimal early-stage traffic windows are sufficient for accurate RAT identification. Pi et al. (2023) demonstrated that byte transition patterns in the first few TCP service packets provide discriminative features for RAT detection with 95.5% accuracy, validating the importance of high-fidelity early-stage capture. Packet-level timestamps were preserved to enable reconstruction of precise event sequences and identification of temporal relationships between network events and host-level activities.

3.5 Feature Extraction and Analysis Techniques

3.5.1 Multi-Dimensional Feature Engineering

Feature extraction encompassed multiple analytical dimensions to capture diverse aspects of AsyncRAT network behavior. The feature engineering pipeline extracted three primary feature categories following established taxonomies in RAT detection research: byte-level features, flow-level statistical features, and behavioral sequence features (Guo et al., 2020; Pi et al., 2023; Zhang et al., 2023).

Byte-level features were extracted from raw packet payloads with particular emphasis on early-stage packets that facilitate rapid detection. Following methodologies established by Pi et al. (2023), byte transition probability matrices were constructed from TCP payload sequences, creating 256×256 representations that capture byte-to-byte transition patterns. These fixed-size matrix representations enable application of convolutional neural network architectures while remaining robust to payload encryption and obfuscation, as they focus on structural transition patterns rather than semantic payload content.

Flow-level statistical features captured aggregate characteristics of network sessions, including packet size distributions, inter-arrival time statistics, flow duration, bidirectional byte counts, and protocol-specific attributes. These features provide complementary information to byte-level representations and have demonstrated efficacy in detecting encrypted RAT communications where payload inspection is limited (Zhang et al., 2023). Statistical features were computed using sliding windows to capture temporal evolution of flow characteristics throughout connection lifecycles.

Behavioral sequence features modeled temporal patterns in network communications, extracting sequences of packet sizes, timing intervals, and protocol state transitions. Sliding-window approaches were employed to segment continuous traffic streams into analyzable sequences, following automated feature extraction methodologies demonstrated in signature generation research (Li et al., 2024). These sequence representations capture the interactive nature of RAT C2 communications and enable detection of multi-stage progression patterns that manifest across extended temporal windows.

3.5.2 Host-Network Feature Correlation

Comprehensive RAT detection benefits from correlating network and host-level indicators to establish causal relationships between system activities and network communications. The methodology incorporated synchronized analysis of network traffic and host telemetry, temporally aligning ETW-logged events—including process lineage, file system modifications, registry persistence mechanisms, and system API call patterns—with network events to construct integrated behavioral profiles (Guo et al., 2020; Zeng et al., 2022). This hybrid approach enables identification of relationships between specific network communications and corresponding host-level actions, facilitating attribution of network traffic to malware processes and reconstruction of complete attack chains.

Temporal synchronization was achieved through high-precision timestamp correlation, aligning ETW event timestamps with packet capture timestamps to sub-second accuracy. This synchronization enables precise attribution of network connections to initiating processes and correlation of file system or registry modifications with subsequent network communications indicative of data exfiltration or command receipt.

3.6 Analysis and Detection Methodologies

3.6.1 Traffic Analysis and Protocol Dissection

Captured network traffic underwent systematic analysis using both automated tools and manual inspection techniques. Protocol dissection employed Wireshark for interactive examination of packet structures, protocol state machines, and application-layer semantics, while Dshell provided scriptable session-level analysis capabilities for

batch processing of large PCAP datasets (Krych & Acosta, 2020). NetworkMiner was utilized for host-centric analysis, extracting transmitted files, credentials, and session metadata from captured traffic.

Analysis focused on identifying characteristic AsyncRAT communication patterns, including C2 handshake sequences, command-response protocols, data exfiltration channels, and payload staging mechanisms. Particular attention was paid to protocol anomalies, non-standard port usage, and attempts to masquerade malicious traffic as legitimate protocols—common evasion techniques employed by contemporary RATs. Protocol analysis incorporated both signature-based pattern matching and behavioral anomaly detection to identify suspicious communications that deviate from expected protocol specifications.

3.6.2 Forensic Reconstruction and Timeline Analysis

Forensic reconstruction synthesized multi-source telemetry to construct comprehensive attack timelines documenting the complete infection chain from initial compromise through post-exploitation activities. The reconstruction methodology adapted structured forensic frameworks documented in recent ransomware and RAT investigation research, specifically employing principles from the TAARA (Trigger, Acquire, Analysis, Report, and Action) methodology for systematic evidence processing (Kusuma et al., 2021). Timeline construction correlated timestamps from PCAP files, ETW logs, file system metadata, and registry modification records to establish precise sequences of events. Causal relationships between events were inferred through temporal proximity analysis and semantic correlation of related artifacts—for example, linking specific network connections to the processes that initiated them through synchronized ETW and packet capture timestamps (Zeng et al., 2022). This multi-source correlation enables reconstruction of attack narratives that would be incomplete from any single telemetry source, providing comprehensive visibility into multi-stage attack progression.

3.6.3 Command-and-Control Infrastructure Characterization

Command-and-control infrastructure identification employed network flow analysis to isolate external communication endpoints contacted by infected hosts. DNS queries, TCP connection establishments, and HTTP/HTTPS sessions were systematically catalogued and analyzed to distinguish legitimate traffic from C2 communications. Indicators such as connection periodicity, data transfer patterns, protocol anomalies, and temporal correlation with host-level malware activities aided in C2 identification (Kusuma et al., 2021; Rao et al., 2024). Identified C2 endpoints were characterized through passive analysis techniques including reverse DNS lookups, WHOIS queries, autonomous system number (ASN) identification, and correlation with threat intelligence databases. Network-level indicators of compromise—including IP addresses, domain names, URL patterns, SSL/TLS certificate characteristics, and HTTP header anomalies—were extracted and documented for operational use in detection systems. This characterization enables development of network signatures suitable for deployment in intrusion detection systems and security information and event management (SIEM) platforms.

3.7 Evaluation Framework and Validation

3.7.1 Detection Performance Metrics

The effectiveness of detection and classification approaches was quantified using standard binary classification metrics including accuracy, precision, recall (True Positive Rate), F1-score, and False Positive Rate, following evaluation practices established in machine learning-based malware detection research (Aburbeian et al., 2025; Guo et al., 2020). These metrics were calculated for both known AsyncRAT variants and previously unseen samples to assess generalization capabilities and robustness against novel variants.

Performance evaluation employed stratified cross-validation to maintain representative distributions of malicious and benign samples across training, validation, and test partitions. Additionally, temporal validation was conducted by training models on earlier-collected samples and testing on more recent variants to evaluate detection durability against evolving malware (Zeng et al., 2022). This temporal validation approach provides more realistic assessment of operational performance than random partitioning, as it simulates the challenge of detecting future malware variants using models trained on historical data.

3.7.2 Forensic Reconstruction Validation

Forensic reconstruction accuracy was validated through comparison with ground truth established during controlled experiments. Since all malware executions occurred in monitored laboratory environments with comprehensive instrumentation, complete knowledge of actual infection sequences was available for validation purposes. Reconstructed timelines and identified C2 infrastructure were systematically compared against known experimental parameters to assess reconstruction completeness and accuracy.

Validation metrics included timeline completeness (percentage of actual events successfully reconstructed), temporal accuracy (precision of event timestamps), and causal relationship accuracy (correctness of inferred event dependencies). These metrics provide quantitative assessment of forensic methodology effectiveness and identify potential gaps in telemetry coverage or analysis techniques (Zeng et al., 2022). Discrepancies between reconstructed and ground truth timelines were analyzed to identify limitations in detection coverage and inform refinement of capture and analysis methodologies.

3.8 Ethical Considerations and Safety Protocols

All malware handling and execution procedures adhered to strict safety protocols to prevent inadvertent system compromise or malware propagation beyond the controlled laboratory environment. The isolated laboratory network maintained complete physical separation from production networks, with multiple layers of containment including network isolation, virtual machine snapshots, and host-based controls (Kusuma et al., 2021; Udeshi et al., 2025). All researchers involved in malware handling received appropriate training in safe malware analysis practices and operational security procedures.

Sample acquisition from public repositories complied with terms of service and ethical guidelines for malware research. No active exploitation of systems or unauthorized access to networks was conducted; all analysis focused exclusively on controlled laboratory environments with researcher-owned infrastructure. Research procedures were designed to advance defensive capabilities and threat understanding without creating risks to external systems, organizations, or individuals. Data management practices ensured that malware samples and captured traffic containing potentially sensitive information remained securely stored with appropriate access controls.

3.9 Reproducibility and Data Management

To facilitate reproducibility and independent validation of findings, comprehensive documentation of experimental procedures, configurations, and parameters was maintained throughout the research process. Virtual machine configurations, network topologies, capture filter specifications, and analysis scripts were systematically documented and archived. Sample cryptographic hash values and metadata were recorded to enable verification of analyzed specimens and replication of experiments by independent researchers.

Raw data artifacts—including PCAP files, ETW logs, and extracted features—were preserved in structured repositories with associated metadata documenting experimental conditions, timestamps, and sample identifiers. Analysis code and feature extraction scripts were version-controlled using Git to maintain provenance of analytical procedures and enable tracking of methodological refinements. This comprehensive data management approach enables independent validation of results and facilitates extension of the research by other investigators, contributing to the broader advancement of malware forensics and network security research.

4. Result

This section presents the forensic findings derived from the investigation of a multi-stage AsyncRAT infection chain. The analysis revealed a structured progression comprising three distinct stages: (1) initial payload delivery, (2) host-level execution and persistence, and (3) command and control (C2) channel establishment. Each stage is supported by empirical evidence, including network captures, host artifacts, and configuration data.

4.1 Stage 1: Initial Infection and Payload Retrieval

The initial phase of the AsyncRAT infection sequence commenced with the execution of a .wsf script embedded within an .iso container. This delivery mechanism circumvented Windows security controls by exploiting the absence of Mark-of-the-Web (MOTW) metadata, thereby enabling silent execution without user prompts. Once activated, the script initiated outbound HTTP requests to a designated staging server over TCP port 222, facilitating the retrieval of two additional components: xlm.txt and mdm.jpg. Although the file appeared to be a harmless image, forensic examination identified it as a hidden PowerShell payload intended to facilitate subsequent stages of the compromise, as shown in Table 1 and Figure 1.

Table 1. Initial Infection Artifacts

Component	Filename	SHA256 Hash	Notes
Initial Vector	invoice#5487214847577.iso	be78…8ed5	SO container bypassing MOTW controls
Loader Script	invoice#5487214847577.wsf	39ce...238	Initiates network download

Downloaded File	/xlm.txt	1e9c...1b1d	Likely auxiliary script
Downloaded File	/mdm.jpg	83ba...fe11	Masqueraded PowerShell payload

Table 1 shows that the attacker employed a multi-component delivery strategy, combining container evasion with script-based retrieval. The use of an ISO file enabled silent execution, while the .wsf loader facilitated dynamic payload acquisition. The presence of a PowerShell script disguised as an image file highlights a deliberate obfuscation tactic aimed at bypassing superficial file-type filters.

Figure 1. Wireshark Capture of HTTP GET Request to Staging Server

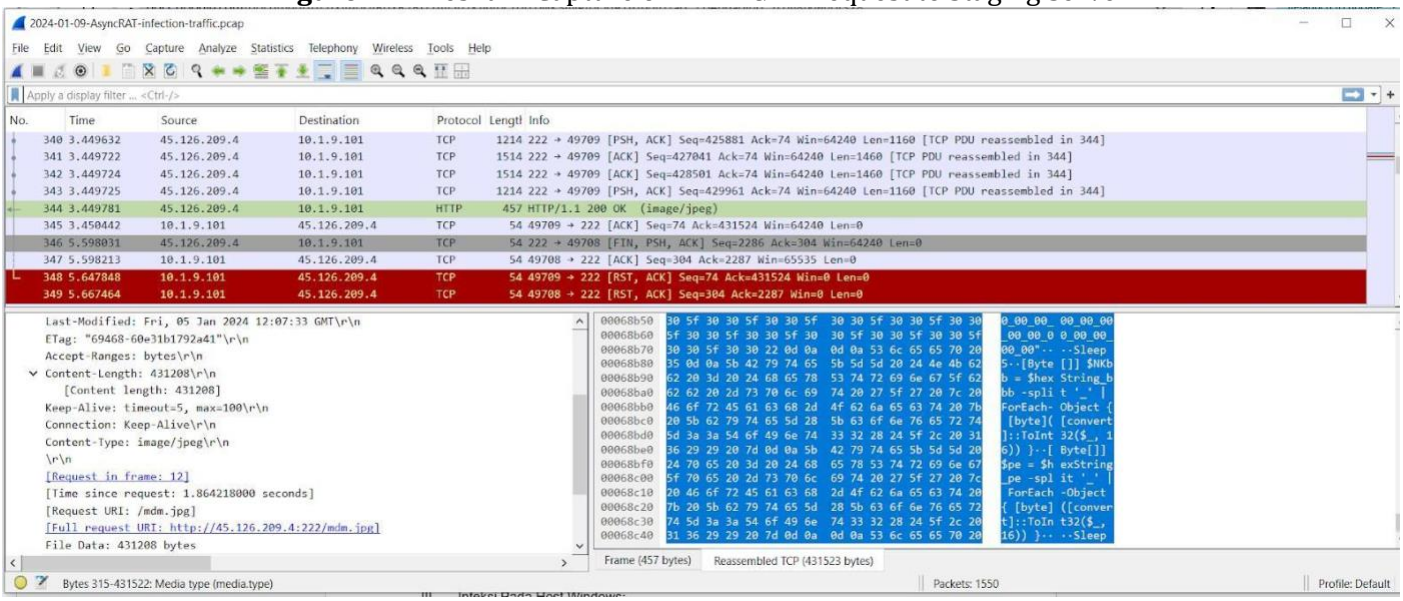


Figure 1 illustrates the obfuscated network behavior during payload retrieval. The HTTP header misrepresents the content type, suggesting an attempt to evade content-aware inspection mechanisms. The use of a non-standard port further complicates detection by conventional firewall and intrusion detection systems. Together, the artifacts in Table 1 and the traffic pattern in Figure 1 demonstrate a well-orchestrated initial infection strategy that combines delivery stealth, dynamic retrieval, and file masquerading to evade both host-based and network-based defenses.

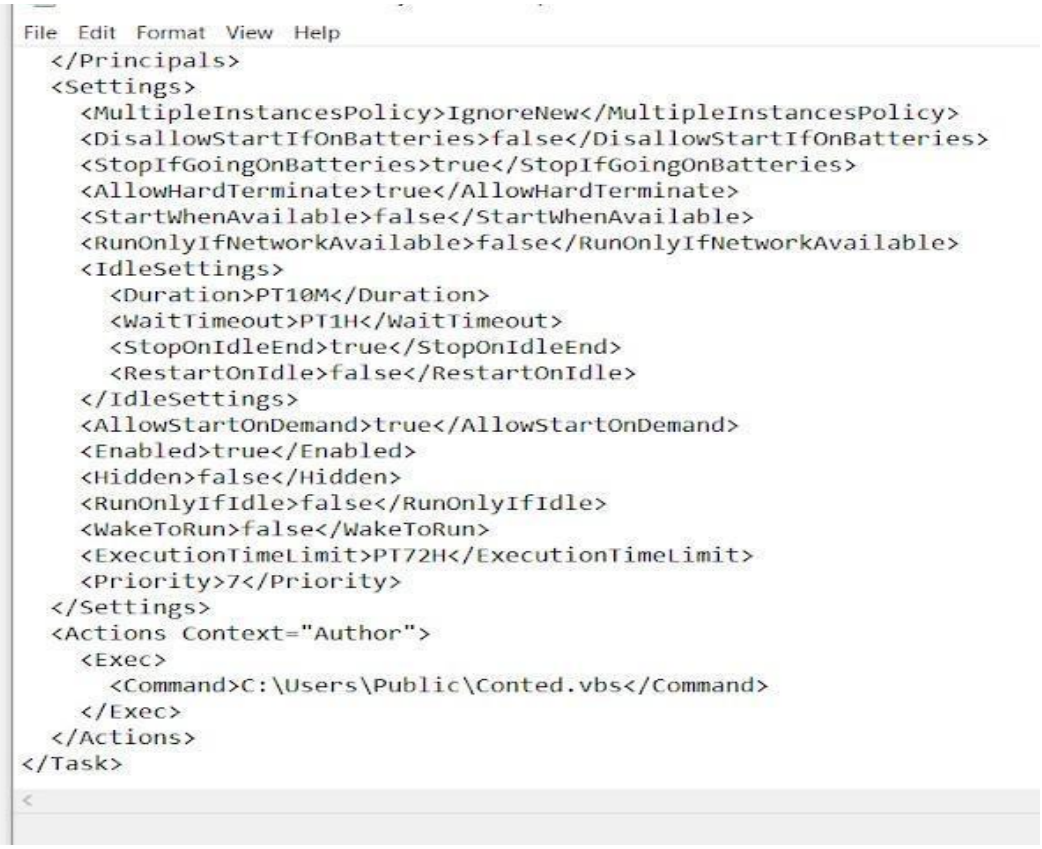
4.2 Stage 2: Host-Level Execution and Persistence

The second stage of the infection chain involved the deployment and execution of multiple scripts designed to maintain persistence on the compromised host. Upon retrieval, the payloads were unpacked into the C:\Users\Public\ directory, a location commonly exploited to evade user scrutiny and facilitate cross-user access. The file mdm.jpg, although bearing an image extension, was in fact a modified PowerShell script. To ensure continued execution of the AsyncRAT payload, the malware utilized three distinct scripts, as detailed in Table 2.

Table 2. Host-Level Artifacts Identified During Forensic Examination

Filename	Type	Size	SHA256 Hash
Conted.bat	Batch Script	205 bytes	cba3...f19
Conted.ps1	PowerShell Script	429 KB	3a0a...593
Conted.vbs	VBScript	688 bytes	a31d...b060

Table 2 shows that the malware employed a multi-script architecture, distributing execution responsibilities across batch, PowerShell, and VBScript formats. This approach increases redundancy and complicates detection by static and behavioral analysis tools. The presence of a large PowerShell script disguised as an image file highlights the use of file masquerading to bypass superficial file-type filters. To maintain persistence, the malware registered a scheduled task configured to execute `Conted.vbs`. The task definition, extracted from the system, revealed permissive execution parameters that allowed the script to run on demand, remain active for up to 72 hours, and operate independently of network availability or idle state, as illustrated in Figure 2.



```
File  Edit  Format  View  Help
  </Principals>
  <Settings>
    <MultipleInstancesPolicy>IgnoreNew</MultipleInstancesPolicy>
    <DisallowStartIfOnBatteries>false</DisallowStartIfOnBatteries>
    <StopIfGoingOnBatteries>true</StopIfGoingOnBatteries>
    <AllowHardTerminate>true</AllowHardTerminate>
    <StartWhenAvailable>false</StartWhenAvailable>
    <RunOnlyIfNetworkAvailable>false</RunOnlyIfNetworkAvailable>
    <IdleSettings>
      <Duration>PT10M</Duration>
      <WaitTimeout>PT1H</WaitTimeout>
      <StopOnIdleEnd>true</StopOnIdleEnd>
      <RestartOnIdle>false</RestartOnIdle>
    </IdleSettings>
    <AllowStartOnDemand>true</AllowStartOnDemand>
    <Enabled>true</Enabled>
    <Hidden>false</Hidden>
    <RunOnlyIfIdle>false</RunOnlyIfIdle>
    <WakeToRun>false</WakeToRun>
    <ExecutionTimeLimit>PT72H</ExecutionTimeLimit>
    <Priority>7</Priority>
  </Settings>
  <Actions Context="Author">
    <Exec>
      <Command>C:\Users\Public\Conted.vbs</Command>
    </Exec>
  </Actions>
</Task>
```

Figure 2. Scheduled Task Configuration for Persistent Execution

Figure 2 illustrates the resilience of the scheduled task configuration. The task is designed to ignore battery constraints, bypass idle restrictions, and remain hidden from standard user interfaces. These settings collectively ensure uninterrupted execution and reduce the likelihood of user detection or system interference. This stage demonstrates the attacker's strategic use of Windows Task Scheduler and scripting diversity to establish durable persistence. The combination of concealed payloads and permissive scheduling parameters reflects a sophisticated evasion methodology, closing the second phase of the infection chain.

4.3 Stage 3: Command and Control (C2) Communication3

The final stage of the infection chain involved the establishment of a persistent outbound connection to the attacker's command and control infrastructure. This connection was initiated by the PowerShell payload and maintained through encrypted communication with a dynamic DNS domain over a non-standard port, as detailed in Table 3 and illustrated in Figure 3.

Table 3. C2 Communication Parameters

Parameterr	Value
C2 Domain	madmrx.duckdns.org
C2 Port	8808 (TCP)
Protocol	HTTPS (TLSv1.0)

Table 3 shows that the malware leveraged madmrx.duckdns.org as its C2 endpoint, utilizing TCP port 8808 and HTTPS encrypted with TLSv1.0. The use of DuckDNS enabled dynamic IP rotation while preserving domain continuity, thereby enhancing operational resilience. The selection of TLSv1.0—an outdated and less scrutinized protocol—suggests a deliberate attempt to evade deep packet inspection systems that prioritize modern TLS versions.

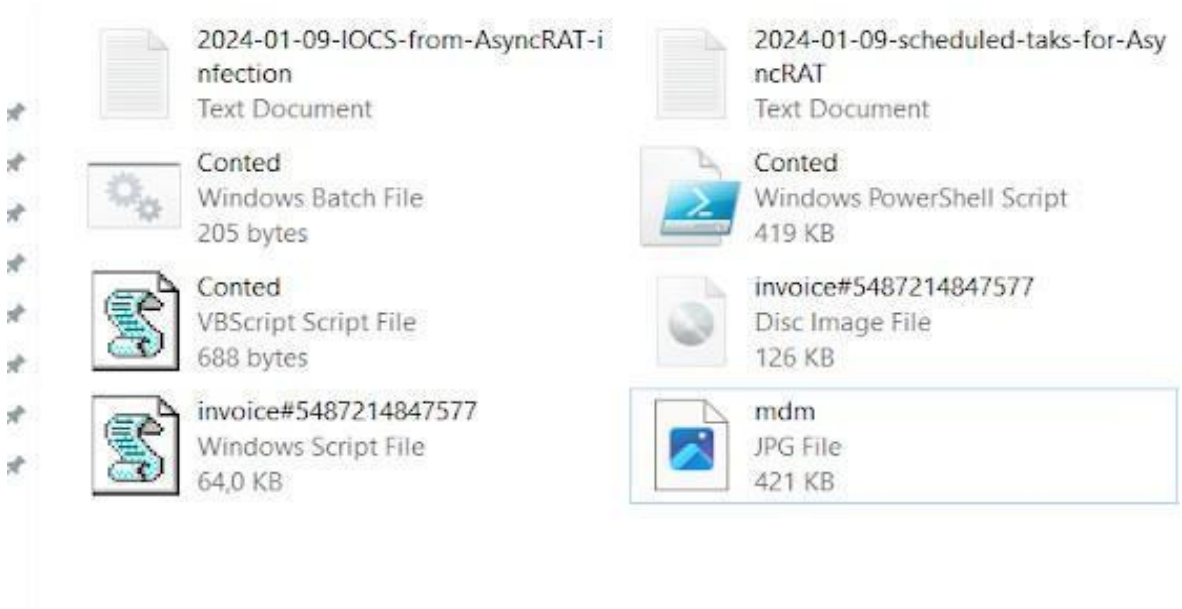


Figure 3. Directory Structure of AsyncRAT Deployment Artifacts

Figure 3 illustrates the complete set of artifacts deployed during the infection process. The presence of multiple script types and a masqueraded payload confirms a layered execution strategy. The directory also includes documentation files, suggesting that the attacker maintained operational notes or indicators of compromise (IoCs) for reuse or reporting. Together, the metadata in Table 3 and the file structure in Figure 3 highlight a high-fidelity behavioral signature. This combination of encrypted communication, dynamic DNS, and non-standard port usage deviates from typical endpoint profiles and can be leveraged for network-based anomaly detection. The persistence of the connection and concealment of payload content via TLS encryption further complicate forensic inspection, underscoring the importance of metadata analysis in identifying covert malware activity.

4.4 Indicators of Compromise and Detection Signatures

The forensic examination of network telemetry revealed several indicators of compromise (IoCs) with strong potential for detection. The infection chain demonstrated a consistent communication pattern, beginning with an outbound connection to a hardcoded IP address (45.126.209.4) over a non-standard TCP port (222). This was subsequently followed by encrypted traffic directed to a dynamic DNS domain (madmrx.duckdns.org) on port 8808, utilizing the outdated TLSv1.0 protocol. Although the use of TLS encryption obscures the content of the communication, the surrounding metadata—such as destination domain, port number, and protocol version—remains observable. These elements provide a distinctive behavioral signature that can be operationalized into intrusion detection rules and network monitoring heuristics. The combination of a non-standard port, deprecated encryption protocol, and dynamic DNS infrastructure constitutes a reliable detection pattern. These findings highlight the critical role of metadata analysis in identifying stealthy malware communications, demonstrating that even in the absence of decrypted payloads, network anomalies can serve as high-fidelity indicators for threat detection.

4.5 Discussion

The observed infection chain demonstrates several modern evasion tactics. The use of an ISO file can bypass Mark-of-the-Web (MOTW) controls in Windows, which would otherwise warn users about executing files from the internet. The multi-stage download process, where a simple script fetches the main payloads, makes static analysis of the initial file more difficult. The network communication is particularly revealing. The initial download from a hardcoded IP on a non-standard port (222) is a strong indicator of malicious activity. The final C2 communication to a duckdns.org domain over port 8808 using TLSv1.0 provides a high-fidelity signature for detection. While TLS encrypts the content of the communication, the metadata—destination domain, port, and TLS version—is highly suspicious and can be used to create effective network security rules. Based on this analysis, the following Indicators of Compromise (IoCs) were identified.

5. Conclusion and Suggestion

This study has conducted a comprehensive forensic analysis of a multi-stage AsyncRAT infection chain, revealing a methodical and evasive attack architecture. The infection sequence progressed through three distinct phases: initial payload delivery, host-level execution and persistence, and command and control (C2) communication. Each phase demonstrated deliberate techniques designed to bypass conventional security mechanisms and complicate forensic investigation. The initial infection leveraged container-based evasion, exploiting the absence of Mark-of-the-Web attributes to silently execute embedded scripts. The use of dynamic payload retrieval over non-standard ports further reduced the visibility of malicious activity during early-stage compromise.

At the host level, the deployment of multiple script types—batch, PowerShell, and VBScript—enabled flexible execution and durable persistence. The configuration of scheduled tasks with permissive runtime parameters ensured long-term operation while minimizing user awareness and system interference. The final stage established encrypted outbound communication with a dynamic DNS endpoint using outdated TLS protocols. This combination of legacy encryption, domain agility, and non-standard port usage produced a high-fidelity behavioral signature that can be leveraged for network-based detection. Overall, the findings underscore the importance of correlating host artifacts with network telemetry to uncover stealthy malware operations. The AsyncRAT campaign exemplifies how modern threat actors integrate obfuscation, modularity, and persistence to maintain control over compromised systems. These insights contribute to the development of more effective detection strategies and reinforce the critical role of metadata analysis in contemporary network forensics.

Reference

Aburbeian, A. H. M., Fernández-Veiga, M., & Hasasneh, A. (2025). Improving remote access trojans detection: A comprehensive approach using machine learning and hybrid feature engineering. AI, 6(9), 237. https://doi.org/10.3390/ai6090237

BehradFar, M. M., HaddadPajouh, H., Dehghantanha, A., Parizi, R. M., Hammoudeh, M., & Choo, K. K. R. (2020). RAT Hunter: Building robust models for detecting remote access trojans based on optimum hybrid features. In Handbook of Big Data Privacy (pp. 335–355). Springer. https://doi.org/10.1007/978-3-030-38557-6_18

Guo, C., Song, Z., Ping, Y., Wang, M., Zhao, J., & Zhao, Y. (2020). PRATD: A phased remote access trojan detection method with double-sided features. Electronics, 9(11), 1894. https://doi.org/10.3390/ELECTRONICS9111894

Hossain, M. N., Milajerdi, S. M., Wang, J., Eshete, B., Gjomemo, R., Sekar, R., Stoller, S. D., & Venkatakrishnan, V. N. (2020). SLEUTH: Real-time attack scenario reconstruction from COTS audit data. In Proceedings of the USENIX Security Symposium (pp. 487–504).

Krych, D. E., & Acosta, J. C. (2020). Hands-on cybersecurity studies: Uncovering and decoding malware communications with Dshell. In Proceedings of the ACM Conference on Innovation and Technology in Computer Science Education (pp. 234–239).

Kusuma, R. S., Umar, R., & Riadi, I. (2021). Network forensics against Ryuk ransomware using Trigger, Acquire, Analysis, Report, and Action (TAARA) method. KINETIK: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 6(2), 131–140. https://doi.org/10.22219/KINETIK.V6I2.1225

Li, X., Yan, H., Lang, B., & Guo, C. (2024). Parallel-SWSA: Automated extraction for feature sequences from remote access trojan attack packets. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 3456–3463). https://doi.org/10.1109/smc54092.2024.10830957

Pi, B., Guo, C., Cui, Y., & Xu, K. (2023). Remote access trojan traffic early detection method based on Markov matrices and deep learning. Computers & Security, 134, 103628. https://doi.org/10.1016/j.cose.2023.103628

Rao, S. S., Savant, R. V., Mishra, S., & Raghunathan, S. (2024). Dissecting digital dangers: A MITRE-aligned analysis of two prominent malware instances. In Proceedings of the International Conference on Computing

Communication and Networking Technologies (ICCCNT) (pp. 1–6). https://doi.org/10.1109/icccnt61001.2024.10725644

Udeshi, M., Putrevu, V. S. C., Krishnamurthy, P., & Khorrami, F. (2025). SaMOSA: Sandbox for malware orchestration and side-channel analysis. arXiv (arXiv:2508.14261). https://doi.org/10.48550/arxiv.2508.14261

Wilkens, F., Ortmann, F., Haas, S., & Fischer, M. (2021). Multi-stage attack detection via kill chain state machines. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS) (pp. 2584–2586). https://doi.org/10.1145/3474374.3486918

Zeng, J., Lin, Z., Zhang, X., Xu, D., & Cheng, Y. (2022). RATScope: Recording and reconstructing missing RAT semantic behaviors for forensic analysis on Windows. IEEE Transactions on Dependable and Secure Computing, 19(3), 1999–2017. https://doi.org/10.1109/TDSC.2020.3032570

Zhang, Y., Han, X., Lin, J., Li, Q., & Shi, J. (2023). ER-ERT: A method of ensemble representation learning of encrypted RAT traffic. In Proceedings of the IFIP Networking Conference (IFIP Networking) (pp. 1–9). https://doi.org/10.23919/ifipnetworking57963.2023.10186391