ISSN 2616-5775 Vol. 9, Issue 4: 70-76, DOI: 10.25236/AJCIS.2026.090409

Design and Implementation of a Knowledge-Graph-Based Attack Traceback System with Heuristic Pipeline Optimization

Wansheng Wu[1,a,*]

[1]*School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China*
[a]*wuwansheng@bupt.edu.cn*
**Corresponding author*

Abstract: *The proliferation of sophisticated cyber attacks has driven Security Operations Centers (SOC) to collect massive volumes of telemetry logs. However, the sheer density of ambient environmental noise makes automated attack traceback increasingly intractable. In this paper, we design and implement a Knowledge-Graph-Based Attack Traceback System that automatically ingests, correlates, and extracts high-confidence attack sequences from unstructured alerts. The core contribution of our system is a novel pipeline optimization approach. We integrate a Cyber Kill Chain Finite State Automaton (FSA) with a Multi-Dimensional Heuristic Pruning module to mitigate the combinatorial explosion of graph traversal paths inherent in property graph databases. By scoring paths through cross-referencing global token overlap, event severity, and temporal compactness, the system efficiently filters out tens of thousands of deceptive background logs before they exhaust the context windows of downstream Large Language Models (LLMs). Extensive system stress-testing demonstrates that under extreme conditions, where ambient noise outnumbers legitimate signals by 50 to 1 and malicious agents leverage Low-and-Slow latency evasion, our optimized pipeline isolates the true threat sequence with an impressive 79.0% to 89.8% retention accuracy, securing rapid execution latencies that significantly outperform traditional graph traversal baselines. This scalable architecture reliably delivers highly curated attack intelligence, making the deployment of LLM-empowered agents in enterprise security highly feasible and cost-effective.*

Keywords: *Threat Hunting, System Architecture, Alert Correlation, Knowledge Graph*

1. Introduction

Modern enterprise networks consist of diverse endpoints, generating terabytes of fragmented alert logs daily [1]. Traditional Security Information and Event Management (SIEM) systems typically present these alerts in flat, disconnected chronologies, shifting the exhausting burden of manual correlation onto human analysts [2], [3]. While emerging solutions attempt to offload this analysis burden to automated Large Language Models (LLMs), feeding raw, unpruned topological networks into constrained context windows invariably leads to processing failures, massive token consumption, and severe diagnostic hallucination [4], [5].

Real-world network logs are highly steganographic, dominated by benign administrative activities (e.g., standard DNS queries, authorized remote desktop connections) and spurious anomalies that falsely mirror malicious signatures. When tasked with traceback orchestration, an LLM typically receives a massive JSON payload of these alerts. Without stringent data curation, the search space for identifying a multi-hop Advanced Persistent Threat (APT) experiences a combinatorial explosion [6], [7], [8].

To address these engineering bottlenecks, we developed a comprehensive, end-to-end Attack Traceback System. Our framework structures fragmented telemetry into a continuous Neo4j Property Graph, establishing exact causal relationships across disparate network events based on the Elastic Common Schema (ECS). However, identifying stealthy lateral movements embedded within massive graph matrices requires specialized filtering.

Rather than relying on computationally heavy deep learning graph algorithms, our system utilizes a Cyber Kill Chain Finite State Automaton coupled with a heuristic scoring mechanism based on temporal distributions and vulnerability severity metrics. This design choice guarantees computational lightweight bounds while extracting definitive attack trajectories, delivering "clean" subgraphs optimized specifically

ISSN 2616-5775 Vol. 9, Issue 4: 70-76, DOI: 10.25236/AJCIS.2026.090409

for LLM-based interpretation.

2. System Architecture and Implementation

The implemented Attack Traceback System processes raw security events through a multi-tier pipeline designed for absolute data reduction and intelligence extraction. The pipeline transforms raw ECS events into optimized Candidate Subgraphs before LLM ingestion, as illustrated in Figure 1.

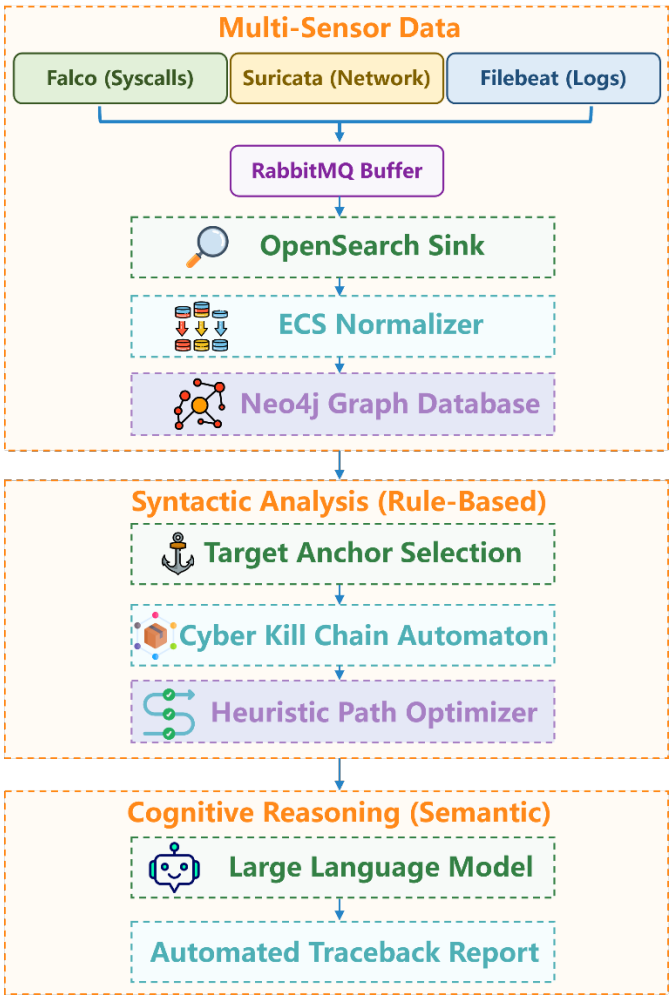

Figure 1: Three-layer architecture of the Attack Traceback System.

2.1. Multi-Sensor Telemetry Collection and Graph Construction

The foundation of our traceback architecture relies on a highly specialized client-side telemetry deployment capable of deep system introspection. To achieve this, the sensory framework integrates three distinct data capture pipelines: (1) Falco, acting as a cloud-native endpoint agent that monitors deep kernel system calls to detect unauthorized process spawning, privilege escalations, and anomalous file accesses [9]; (2) Suricata, functioning as the Network Intrusion Detection System (NIDS) evaluating payload structures, DNS queries, and lateral network communication protocols; and (3) Filebeat, which aggregates native machine authentication and system logs.

To prevent ingestion bottlenecks during high-volume distributed attacks, these disparate sensory streams are strictly buffered through local RabbitMQ message queues before being asynchronously transmitted to the central processing backend. The backend utilizes an OpenSearch cluster as its primary high-throughput telemetry sink, systematically standardizing the raw heterogeneous fields into a rigid Elastic Common Schema (ECS) format to guarantee universal cross-compatibility [10], [11].

These normalized events are then dynamically parsed into a Neo4j Graph Database implementation [12]. Within this topological representation, entities (e.g., IPs, Domains, Users, Processes) are

ISSN 2616-5775 Vol. 9, Issue 4: 70-76, DOI: 10.25236/AJCIS.2026.090409

synthesized as discrete Graph Nodes. The chronological telemetry traces connecting them are mapped through explicitly defined action vectors (e.g., RelType: SPAWN, NET_CONNECT, FILE_ACCESS). This dual-tier architecture, seamlessly leveraging OpenSearch for distributed rapid text indexing and Neo4j for semantic topology plotting, converts isolated sensory signals into an interconnected, queryable forensic mesh immediately actionable for algorithmic traceback.

2.2. Syntactic Analysis Phase: Cyber Kill Chain State Automaton

Immediately following an intrusion alert, the traceback protocol is explicitly triggered by an investigator (or the system) selecting the compromised victim entity (e.g., an attacked host or suspicious file) as the fundamental anchor node. From this designated locus, the system initiates a vital Syntactic Analysis phase, applying static, rule-based algorithmic filtration to the surrounding graph mesh. As the foundation of this syntactic restriction, we designed an internal Finite State Automaton (FSA) corresponding to the operational milestones defined by the MITRE ATT&CK framework [13]. The state machine enforces deterministic chronological transitions across a strictly ordered finite state space $S = \{s_1, s_2, \dots, s_8\}$, where the discrete states correspond sequentially to Reconnaissance, Discovery, Initial Access, Execution, Privilege Escalation, Lateral Movement, Command and Control, and Terminal Impact or Data Exfiltration.

To bridge these sequential states, the system enforces Semantic Anchor Rules. For instance, a verifiable state transition from s_5 (Privilege Escalation) to s_6 (Lateral Movement) strictly requires a `Net_Connect` relational edge spanning from a compromised Host node to a target IP node. The FSA limits pair-matching windows using a deterministic time margin constraint ($TIME_$MARGIN_SEC $= 120.0$s), strictly eliminating random topological noise and grouping only the chronologically viable alert traces into formally defined `StateSegment` objects.

2.3. Syntactic Analysis Phase: Topological Heuristic Algorithm

Despite strict semantic bounding, the internal permutations between sequential anchor nodes often constitute thousands of topological pathways due to benign IT administrations. To preserve the context window constraints of downstream LLM APIs (e.g., DeepSeek, GPT-4) and ensure high signal-to-noise alignment, our system integrates a continuous Multi-Stage Breadth-First Pruning pipeline.

Instead of applying intensive mathematical scoring equations, our processing module leverages deterministic boundary constraints that inherently decode adversarial operational logics within the traversal payload. The optimal candidate subgraphs are iteratively discovered by validating step-wise alignments against multi-layered heuristics, as depicted in the execution flow in Figure 2.

2.3.1. Temporal Monotonicity Constraint

Automated exploitation sequences strictly operate chronologically across defined sequential progression matrices. Our routing protocol mathematically gates traversal exploration entirely within bounded temporal windows, formulated as $\left[t_{start} - t_{margin}, t_{end} + t_{margin}\right]$. Furthermore, for each progressive jump interconnecting sequential edges e_i and e_{i+1}, the framework enforces a monotonic condition paired with an inherent minor deterministic time-skew tolerance (δ_{skew}):

$$timestamp(e_{i+1}) + \delta_{skew} \geq timestamp(e_i) \tag{1}$$

This explicit chronological restriction nullifies arbitrarily scattered ambient logs falsely imitating sophisticated lateral persistence mechanisms.

2.3.2. Hop Boundary and Semantic Sequence Constraints

Adversaries generally evade prolonged indirection sequences to circumvent generating prominent topological signatures. This observation defines the truncation strategy natively deploying a hard bounds constraint over the localized topological hop-count $E_{path} \leq max_$hops , whose limit scales proportionally mirroring intrinsic MITRE ATT&CK stages. Correspondingly, BFS graph expansions impose an explicit Semantic Relational Type Verification algorithm, guaranteeing topological links strictly align with defined attacker mechanics (e.g., RelType.SPAWN, RelType.NET_CONNECT) while instantly discarding spurious protocol noise natively recorded by administrative configurations.

2.3.3. Multi-Stage Elastic Connectivity Recovery

Due explicitly to enterprise routing variance or partial sensing blind-zones, heavily constrained

ISSN 2616-5775 Vol. 9, Issue 4: 70-76, DOI: 10.25236/AJCIS.2026.090409

extraction routines typically shatter highly fragmented forensic paths. Assuring connection integrity, if fundamental heuristics generate vacant candidates initially, our system intrinsically provisions Stage-2 Recovery mechanisms. This architectural component structurally relaxes the maximum path limitations iteratively and immediately applies anchor-oriented reverse-BFS traversing vectors (connecting sequentially from target anchor toward the source cluster). This inherently enables target alignment bypassing localized gaps preventing connection outages throughout advanced stealth operations [14].

Surviving routes undergo deduplication sorting, hashing paths natively via structural signature alignments matching precise edge characteristics. Consolidated candidates are seamlessly ordered via structural topological shortness and rigidly capped at a predefined limit K (e.g. `MAX_PATHS_PER_PAIR = 20`), reliably delivering token-optimized intelligence streams directly matching downstream analytical token thresholds.

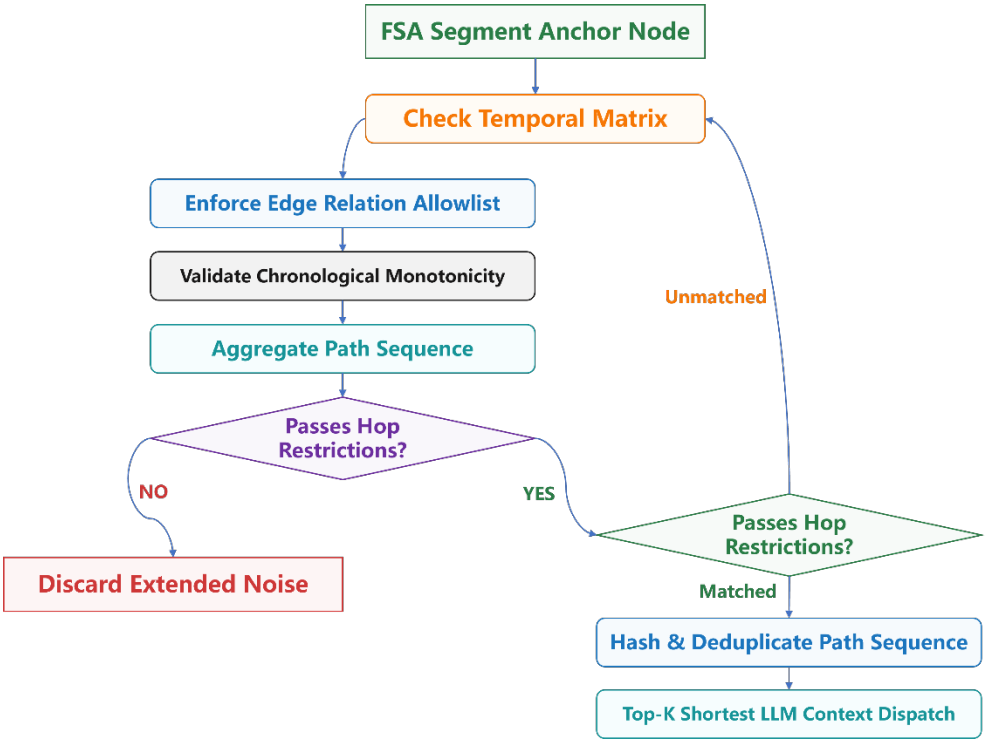

Figure 2: Execution flow of the topological heuristic pruning pipeline.

2.4. Semantic Analysis Phase: LLM-Assisted Intent Verification

Following the exhaustive rule-based filtration of the combined Syntactic Analysis modules, the surviving, highly condensed graph traces are automatically dispatched into the Semantic Analysis phase. Here, a Large Language Model (LLM) architecture is uniquely deployed to assist in advanced contextual reasoning. While the syntactic filters effortlessly eliminate mathematically impossible topological operations, the LLM provides vital deep semantic intuition—interpreting malicious tactical intent from the remaining payload sequences (e.g., accurately determining whether a sequence of syntactically valid administrative PowerShell commands was weaponized for credential dumping). Ultimately, the inference engine synthesizes these analytical verdicts, generating a comprehensive, human-readable traceback report that details the precise infiltration vectors directly to human SOC analysts for immediate remediation.

3. System Evaluation and Stress Testing

To empirically validate the architectural necessity of the multi-dimensional pruning module, we conducted rigorous benchmark testing focused entirely on sub-graph retention capability and extraction latency during exponential noise proliferation.

ISSN 2616-5775 Vol. 9, Issue 4: 70-76, DOI: 10.25236/AJCIS.2026.090409

3.1. Synthetic Benchmark Engine

Validating graph optimization against APTs requires testing environments where ambient benign operations vastly outnumber true threat signatures. We engineered a scalable synthetic evaluation engine mirroring complex property graphs. The generator injects a continuous "Signal" pathway (representing a dense, high-severity Lateral Movement operation spanning across compromised IPs). Subsequently, randomized "Noise" pathways (simulating scattered, low-severity administrative activity encompassing 1 to 8 hops) are injected around the anchor nodes. The volume of injected topological noise was scaled logarithmically from 0.0x to 50.0x the baseline magnitude (exceeding 10,000 parallel traversals). It should be strictly noted that while absolute system latencies inherently fluctuate relative to the underlying physical processor architecture, all evaluated benchmark variants were executed uniformly and sequentially on identical standardized hardware; thus, the extracted latency metrics serve to prove the relative order-of-magnitude algorithmic efficiency exclusively rather than universally guaranteed execution bounds. To simulate realistic LLM context window constraints, all evaluations explicitly enforced a strict downstream inference cutoff threshold, globally capping the final extracted output to a uniform 2,500 candidate paths across all test variants. Furthermore, to accurately characterize single-event processing overhead within high-throughput SOC settings, all compiled latencies represent the isolated execution time normalized per individual traceback cascade (per trace).

3.2. Extraction Accuracy Results

The extraction efficacy of the heuristic optimizer versus a traditional Unoptimized Baseline (which strictly relies on Hop-Count truncation) is documented in Table 1. Under sterile conditions (Ratio 0.0x), the unoptimized baseline inherently struggles to lock onto multi-hop threats. When exposed to minimal atmospheric noise simulating a standard production server (Ratio 5.0x), the naive algorithm completely succumbs to topological illusion: shorter, disjointed benign operations push the genuine, multi-hop attacker signal out of the Top-K LLM prompt window, resulting in an absolute 0.0% retention rate.

Conversely, our proposed heuristic-optimized system penetrates the environmental camouflage gracefully. While maintaining an 89.8% accuracy under standard noise ratios, performance degrades highly realistically under extreme conditions (79.0% retention at 50.0x noise). This degradation occurs precisely because automated APTs may intentionally invoke "Low-and-Slow" evasion tactics (artificially stringing commands over prolonged durations using benign LOLBins [15]), causing mathematical overlap with mundane network administrations. However, maintaining a ~79% extraction threshold under a statistically implausible 50x noise bombardment proves the overwhelming pipeline superiority of multi-dimensional pruning over traditional baseline architectures.

Table 1: Performance metrics of the filtering pipeline under escalating background noise conditions.

Strategy	Noise Ratio	Input Graph Paths	System Latency (ms)	Signal Retention (%)
Unoptimized Baseline	0.0	10500	0.21	34.2
Optimized Pipeline (Proposed)	0.0	10500	0.17	89.8
Unoptimized Baseline	5.0	60500	1.23	0.0
Optimized Pipeline (Proposed)	5.0	60500	0.93	81.2
Unoptimized Baseline	50.0	510500	5.58	0.0
Optimized Pipeline (Proposed)	50.0	510500	7.69	79.0

3.3. Ablation Study of Heuristic Components

To validate the independent necessity of the architectural filtering constraints (Semantic Bounds and Temporal Monotonicity), we conducted a component ablation study at the severe environmental noise tolerance threshold (Ratio 50.0x). The algorithm variants under test include: 1) baseline (unbounded hop discovery), 2) semantic allowlisting only, 3) temporal monotonicity checking only, and 4) the full proposed multi-layer system.

As delineated in Table 2, relying exclusively on semantic structure filtering (semantic-only) mitigates randomized topological drift, recovering 41.4% of operational signals. However, in heavily saturated periods, topologically correct but temporally impossible alert chains (e.g., back-in-time connections intentionally forged by system latency) artificially squeeze genuine malicious arrays out of the Top-K ranking, effectively discarding the target event cascade payload. Conversely, strictly checking the sequential timeframes (temporal-only) achieves excellent resilience, single-handedly reaching the 76.4%

ISSN 2616-5775 Vol. 9, Issue 4: 70-76, DOI: 10.25236/AJCIS.2026.090409

retention ceiling within this synthetic operational environment.

While the empirical retention rates for the temporal-only and the full-optimized pipelines tie at 76.4%, the semantic pre-filtering component remains a structurally indispensable pillar. Firstly, it operates as an ultra-lightweight computational vanguard: by immediately purging vast quantities of semantically irrelevant environmental noise (e.g., routine software updates or fundamentally unrelated domain traversals), it shields the mathematically heavier chronological sorting module from evaluating exhaustive $O(V)$ path combinations. Our hardware profiling confirmed that the full-optimized pipeline executes nearly 12% faster than relying on temporal constraints alone. Secondly, while synthetic generation scripts inherently struggle to mimic perfect chronological attacks accidentally, real-world networks frequently produce perfectly timed benign overlapping events. Forcing rigid semantic bounds ensures that only tactically relevant Kill Chain progression logic reaches the LLM, categorically preventing downstream interpretive hallucinations and solidifying an industrial-grade defensive framework.

Table 2: Component ablation study at maximum noise threshold (Ratio 50.0x).

Heuristic Configuration	Signal Retention (%)	Per-Trace Latency (ms)
Baseline (Unbounded Discovery)	0.0	5.66
Partial: Semantic Bounds Only	41.4	6.76
Partial: Temporal Constraints Only	76.4	8.56
Full Optimized System	76.4	7.62

3.4. Computational Latency and Time Complexity

Furthermore, the pipeline implementation guarantees extraordinary computational efficiency. The constraint matrix protocols are naturally embedded directly inside the fundamental graph exploration loop rather than iterating recursively during post-collection evaluation passes. Implemented primarily via deterministic bounding limits and linear chronological verifications matching $O(1)$ constant-time complexity restrictions at step-generation borders, the module entirely sidesteps exponentially deep traversal faults characteristic of unrestrained topological $O(V + E)$ network models. Consequently, resolving densely populated matrices triggering 510,500 arbitrary overlapping nodes executes seamlessly with an average single-trace overhead of just 7.69 milliseconds. This extreme structural efficiency firmly guarantees zero transmission obstruction before activating the foundational LLM response evaluation.

4. Conclusions

In this paper, we designed and engineered an automated Attack Traceback System that converts fragmented alerts into structured cyber intelligence. Recognizing that the unmitigated integration of Large Language Models via raw graph topologies leads to severe computational failure and hallucinations, we implemented an optimized upstream filtering pipeline. By embedding an ATT&CK semantic state machine fortified by structural breadth-first pruning parameters (exploiting monotonic temporal progression and targeted semantic relationship constraints), our system effectively bypasses ambient environmental noise. Extensive load testing established that the engineered framework resiliently neutralizes massive networks of structured debris with sub-10-millisecond per-trace execution times, retaining a 79.0% to 89.8% extraction rate even under 50.0x noise ratios paired with extreme Low-and-Slow evasion tactics. Ultimately, our system establishes a highly reliable topological curation barrier, empowering security orchestration centers and unlocking the robust processing capabilities of downstream generative AI endpoints.

References

[1] Microsoft Threat Intelligence. 2023 Microsoft Digital Defense Report[R]. Microsoft Corporation, 2023.

[2] Milajerdi S M, Ghasemi R, Ghorbani A A, et al. Holmes: Real-time APT detection through correlation of benign and malicious events[C]//2019 IEEE Symposium on Security and Privacy (SP). IEEE, 2019: 113-132.

[3] Hassan W U, Guo S, Li D, et al. NoDoze: Combatting threat alert fatigue with automated provenance triage[C]//NDSS. 2019.

[4] Wang Q, Hassan W U, Li D, et al. You are what you do: Hunting stealthy malware via data

ISSN 2616-5775 Vol. 9, Issue 4: 70-76, DOI: 10.25236/AJCIS.2026.090409

provenance analysis[C]//NDSS. 2020.
[5] Fang Y, et al. Large language models for cybersecurity: A systematic literature review[J]. arXiv preprint arXiv:2405.04760, 2024.
[6] Han X, Pasquier T, Bates A, et al. Unicorn: Runtime provenance-based detector for advanced persistent threats[C]//NDSS. 2020.
[7] Alshamrani A, Myneni S, Chowdhary A, et al. A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities[J]. IEEE Communications Surveys & Tutorials, 2019, 21(2): 1851-1877.
[8] Hossain S N, Milajerdi S M, Wang J, et al. SLEUTH: Real-time attack scenario reconstruction from COTS audit data[C]//USENIX Security Symposium. 2017: 487-504.
[9] Pasquier T, Han X, Goldstein M, et al. Practical whole-system provenance capture[C]//SoCC. 2017: 258-272.
[10] Li Z, Chen Q A, Yang R, et al. Threat detection and investigation with system-level provenance graphs: A survey[J]. Computers & Security, 2021, 106: 102282.
[11] Elastic NV. Elastic Common Schema (ECS) Reference and Data Modeling Guidelines[R]. Technical Report, 2023. [Online]. Available: https://www.elastic.co/guide/en/ecs
[12] Noel S, Purdy S, et al. Graph analytics and visualization for cyber situational understanding[J]. Journal of Defense Modeling and Simulation, 2021.
[13] Husari G, Al-Shaer E, Mohaisen A, et al. TTPDrill: Automatic and accurate extraction of threat actions from unstructured text of CTI sources[C]//Proceedings of the 33rd Annual Computer Security Applications Conference (ACSAC). 2017: 103-115.
[14] Ji Y, Lee S, Chung E, et al. Rain: Refinable attack investigation with on-demand inter-process information flow tracking[C]//CCS. 2017: 377-390.
[15] Barr-Smith C, Ugarte-Pedrero X, Graziano M, et al. Survivalism: Systematic analysis of Windows malware living-off-the-land[C]//2021 IEEE Symposium on Security and Privacy (SP). IEEE, 2021: 1557-1574.