SCIENCE CHINA
Information Sciences

• LETTER •

September 2026, Vol. 69, Iss. 9, 190311:1–190311:2
https://doi.org/10.1007/s11432-025-4995-0

Special Topic: User-Centric Intelligent Networking

Honey Insight: the interpretable reasoning engine for dynamic TTP inference

Yinghai ZHOU[1,2,3†], Ziyu WANG[1†], Rui WANG[1], Yuyu HE[1*], Hui LU[1,2,3] & Zhihong TIAN[1,2,3*]

[1]*Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China*
[2]*Guangdong Province Key Laboratory of Industrial Control System Security, Guangzhou 510006, China*
[3]*Huangpu Research School of Guangzhou University, Guangzhou 510006, China*

Received 9 October 2025/Revised 1 March 2026/Accepted 5 June 2026/Published online 25 August 2026

Citation Zhou Y H, Wang Z Y, Wang R, et al. Honey Insight: the interpretable reasoning engine for dynamic TTP inference. Sci China Inf Sci, 2026, 69(9): 190311, https://doi.org/10.1007/s11432-025-4995-0

Endpoints and network assets are increasingly targeted by sophisticated attacks such as advanced persistent threats (APTs), where traditional traffic-based detection methods prove inadequate. Deception defense offers a proactive approach by creating realistic decoy environments to mislead attackers, disrupt their paths, and capture behaviors.

The Honeycenter system [1] advances beyond static honeypots by integrating behavior detection, software-defined networking (SDN), container technology, and dynamic analysis, enabling cross-domain deployment, elastic scaling, and multi-service deception. Upon redirection into Honeycenter, malicious actions trigger the behavior detection module.

Honey Insight constitutes the core intelligent innovation of the Honeycenter system. While prior Honeycenter modules focus on SDN-based traffic redirection, container-driven trap-network construction, and Bayesian attack-graph decision-making for proactive deception, they lack real-time attacker strategy inference from behavioral logs. Existing TTP inference methods either rely on black-box deep learning (offering high accuracy but no traceability) or rigid rule-based systems (providing interpretability but limited coverage and adaptability). In contrast, Honey Insight introduces the first fully traceable "Logs → System Behaviors → TTPs" reasoning chain by combining LLM-guided log projection with a two-stage interpretable machine-learning framework (logistic regression + binary association rules). This design enables adaptive, explainable deception that dynamically reconstructs the trap environment according to inferred attacker TTPs under the MITRE ATT&CK framework.

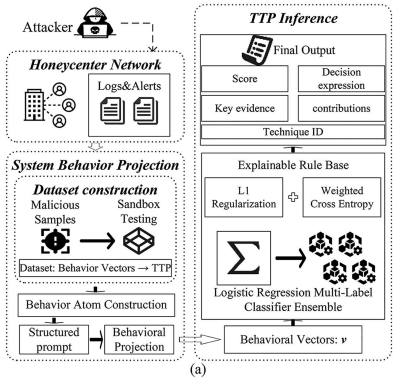
Problem statement. Honey Insight addresses three core challenges to establish the interpretable "Logs → System Behaviors → TTPs" reasoning chain. First, it constructs a reliable data foundation by defining 26 system behavior atoms and building a dataset D mapping malware samples to both atomic behavior frequency vectors v_i and multi-label TTP vectors y_i. Second, it tackles log projection by designing a function f that leverages LLMs with structured prompts to convert unstructured logs X into standardized behavior vectors $\{v_i\}$. Finally, it performs TTP inference by learning an interpretable mapping g from behavior vectors to TTP labels, ensuring transparency and robustness. To address these challenges, Honey Insight proposes the following methodology.

Methodology. Honey Insight addresses the limitations of existing TTP inference methods—the poor interpretability of deep learning models and the limited coverage of rule-based systems—through a two-stage framework. As shown in Figure 1(a), the framework first projects raw logs onto a set of predefined system behavior atoms, then employs interpretable machine learning to map these atomic behaviors to specific TTPs. This design preserves the expressive power of data-driven approaches while ensuring full traceability of decisions.

Dataset construction. A dataset linking system behaviors to TTPs is constructed from 10126 verified malware samples collected from public repositories (VirusTotal) and internal sandbox executions. Each sample was executed in a controlled Cuckoo Sandbox environment, and its traces were manually verified and annotated by cybersecurity experts. After normalization, execution traces are mapped to 26 system behavior atoms. Each sample is represented as a frequency vector $v_i \in \mathbb{N}^{26}$ and associated with a multi-label TTP vector $y_i \in \{0,1\}^{292}$. Following statistical significance filtering (minimum 30 positive samples per label), 253 TTP labels are retained. This approach builds on prior work in cybersecurity knowledge graphs for APT organization attribution [2–4]. The complete list of the 26 system behavior atoms, together with category groupings, frequency statistics, and operation semantic examples, is provided in Appendix A. Detailed statistical distributions of the malware samples (family distribution, platform/OS and TTP categories) are given in Appendix B.

System behavior projection. Unstructured logs are mapped to behavior atom vectors via a large language model guided by structured prompts. The prompt enforces role specification, task description, atom definitions, and JSON output constraints, reducing output randomness. For each log fragment x_i, the model produces a 26-dimensional count vector v_i and a provenance set

* Corresponding author (email: heyy@gzhu.edu.cn, tianzhihong@gzhu.edu.cn)
† These authors contributed equally to this work.



(a)

Model/Log Type	Accuracy (%)	Cost (¥)
deepseek-v3	89.33	0.08
qwen-plus (Structured Prompt)	90.71	0.05
qwen-plus (Simple Instruction)	74.50	0.05
Kimi-K2	87.60	0.12
glm-4.5	61.24	0.15
Honeypoint Logs (Qwen-Plus)	94.20	–
EDR Alert Logs (Qwen-Plus)	87.20	–

(b)

Alg.	P_{macro}	R_{macro}	$F1_{\text{macro, Sandbox}}$	$F1_{\text{macro, Honeycenter}}$	$\Delta F1$
LR	0.6425	0.5981	0.6002	0.6132	-0.0130
RF	0.7267	0.6107	0.6199	0.4647	0.1552
SVM	0.2612	0.5139	0.2741	0.2155	0.0586
NB	0.5561	0.5719	0.3316	0.4103	-0.0787

(c)

Figure 1 (a) Overall framework of the Honey Insight attacker strategy inference system; (b) evaluation results of system behavior projection; (c) evaluation results of TTP inference.

linking each detected atom to the original log entries. This yields normalized behavioral features while preserving interpretability.

TTP inference. TTP prediction is formulated as a multi-label classification problem solved by M independent logistic regression classifiers (one per TTP). The probability for technique t given behavior vector v is $P(y_t = 1 \mid v) = \sigma(\mathbf{w}_t^\top v + b_t)$, where σ is the sigmoid function. To handle class imbalance and feature scale differences, min-max normalization is applied, and a weighted cross-entropy loss with L_1 regularization is optimized. The resulting rule set $\mathcal{R}$ stores for each technique its weight vector, bias, decision threshold, and a mapping table for interpretability.

During inference, a new behavior vector v is scored against all rules; techniques with scores above the threshold are flagged as triggered. For each triggered technique, the system generates a decision path: it computes per-feature contributions $\delta_k = w_{t,k} \cdot v_k$, selects significant features, and outputs a linear expression $b_t + \sum_{k \in F_t} w_{t,k} v_k \geqslant \sigma^{-1}(\theta_t)$. The final output $\mathcal{O}$ includes the technique ID, score, key evidence, contributions, and decision expression, thereby answering not only "which TTPs" but also "why". For example, when the behavior vector shows high frequencies in b_2 (files_written) and b_9 (processes_injected), the system may output: "T1059.003 is triggered because $0.42 \times v_2 + 0.31 \times v_9 + b_t \geqslant 0.65$ (threshold), with files_written contributing 68% of the score." This linear expression and per-feature contribution table enable security analysts to trace every decision back to the original logs.

Evaluation. Log projection. Uses LLMs to map unstructured logs to 26 behavioral atoms. On 600 annotated logs (300 honeypoint [5] + 300 endpoint detection and response (EDR)), four LLMs were compared (structured prompts, JSON output). As shown in Figure 1(b), Qwen-Plus achieves the highest accuracy (90.71%) with the lowest cost (0.05 CNY/call). Structured prompts improve accuracy from 74.5% to 90.7%. Cross-log testing yields 94.2% on honeypoint logs and 87.2% on EDR logs.

TTP inference. One logistic regression (LR) model is trained per ATT&CK technique, using behavioral vectors as input from a dataset of 10126 malware samples. Model comparison employs sandbox test set metrics (macro precision, recall, macro F1) and cross-domain Honeycenter test set metrics (macro F1 and $\Delta F1$).

As shown in Figure 1(c), LR balances performance and interpretability, achieving a macro F1 of 0.6002 on the sandbox test set and 0.6132 on the Honeycenter test set. We define $\Delta F1 = F1_{\text{macro, sandbox}} - F1_{\text{macro, Honeycenter}}$. A positive $\Delta F1$ indicates degradation (overfitting to sandbox patterns), while a negative value (LR: $\Delta F1 = -0.0130$) highlights strong cross-domain generalization. Random forest (RF) yields a higher sandbox macro F1 (0.6199) but suffers a sharp drop ($\Delta F1 = 0.1552$). SVM and Naive Bayes (NB) exhibit poor predictive performance and weak generalization overall.

For fair assessment, uninterpretable baselines XGBoost and a 3-layer multi-layer perceptron (MLP) were also evaluated. XGBoost reaches a sandbox macro F1 of 0.625 but degrades markedly ($\Delta F1 = +0.138$), indicating overfitting. MLP shows similar generalization issues ($\Delta F1 = +0.112$) and lacks feature traceability. Consequently, LR is selected as the optimal solution for Honey Insight, uniquely balancing competitive accuracy, robust cross-domain generalization (negative $\Delta F1$), and full coefficient-level interpretability.

Load performance. Qwen-Plus achieves 5.85 seconds per segment (s/seg) latency, 0.16 seg/s throughput, and 2.82% CPU; LR is lightweight (0.003 s/seg, 337 seg/s, 7.84% CPU), meeting real-time dynamic defense requirements.

Conclusion. Honey Insight plays a pivotal role in strengthening Honeycenter's dynamic deception capabilities. Upon attacker entry into the initial network, honeypoint behavioral logs feed into Honey Insight for real-time TTP inference. The results are relayed to the scheduling module, which dynamically builds tailored trap networks via container technology and redirects traffic through SDN. This iterative process of log collection, strategy inference, environment reconstruction, and redirection continuously adapts to attacker actions, confining them within an evolving deception landscape while enabling full-cycle behavior capture, identity tracing, and attack chain disruption. Overall, Honey Insight delivers an interpretable reasoning engine with high accuracy, low latency, and strong cross-domain generalization in real-world scenarios.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 62302115, U2436208, 62372129), Guangdong S&T Program (Grant No. 2024B0101010002), Guangdong Provincial Key Laboratory of Industrial Control System Security (Grant No. 2024B1212020010), and National Key Research and Development Plan (Grant No. 2023YFB3107300).

Supporting information Appendixes A and B. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Wang R, Liu Y, Sun Y, et al. H^4: a software-defined deception defense system in safeguard defense mode. IEEE Trans Dependable Secure Comput, 2026, 23: 507–524
2 Zhou Y, Wang Z, Jiang Y, et al. AEKG4APT: an AI-enhanced knowledge graph for advanced persistent threats with large language model analysis. ACM Trans Intell Syst Technol, 2025. doi: 10.1145/3735645
3 Wang Z, Zhou Y, Liu H, et al. ThreatInsight: innovating early threat detection through threat-intelligence-driven analysis and attribution. IEEE Trans Knowl Data Eng, 2024, 36: 9388–9402
4 Ren Y, Xiao Y, Zhou Y, et al. CSKG4APT: a cybersecurity knowledge graph for advanced persistent threat organization attribution. IEEE Trans Knowl Data Eng, 2023, 35: 5695–5709
5 Wang R, Yang C, Deng X, et al. Turn the tables: proactive deception defense decision-making based on Bayesian attack graphs and Stackelberg games. Neurocomputing, 2025, 638: 130139