Poster: WATSON: Abstracting Behaviors from Audit Logs via Aggregation of Contextual Semantics

Jun Zeng[†] Zheng Leong Chua[‡+] Yinfang Chen[†] Kaihang Ji[†] Zhenkai Liang[†] Jian Mao[§]
[†]School of Computing, National University of Singapore
[‡]Independent Researcher
[§]School of Cyber Science and Technology, Beihang University

***Abstract*—Endpoint monitoring solutions are widely deployed in today's enterprise environments to support advanced attack detection and investigation. These monitors continuously record system-level activities as audit logs and provide deep visibility into security incidents. Unfortunately, to recognize behaviors of interest and detect potential threats, cyber analysts face a semantic gap between low-level audit events and high-level system behaviors. To bridge this gap, existing work largely matches streams of audit logs against a knowledge base of rules that describe behaviors. However, specifying such rules heavily relies on expert knowledge. In this paper, we present WATSON, an automated approach to abstracting behaviors by inferring and aggregating the semantics of audit events. WATSON uncovers the semantics of events through their usage context in audit logs. By extracting behaviors as connected system operations, WATSON then combines event semantics as the representation of behaviors. To reduce analysis workload, WATSON further clusters semantically similar behaviors and distinguishes the representatives for analyst investigation. In our evaluation against both benign and malicious behaviors, WATSON exhibits high accuracy for behavior abstraction. Moreover, WATSON can reduce analysis workload by two orders of magnitude for attack investigation. We present an end-to-end design and support for audit event analysis on multiple system platforms.**

I. INTRODUCTION

Security incidents in large enterprise systems have been on the rise globally. We have been witnessing attacks with increasing scale and sophistication. For example, Capital One reported that 106 million customers' credit card information was exposed due to unauthorized database access. To better prevent and respond to such attacks, endpoint monitoring solutions (e.g., Security Information and Event Management (SIEM) tools) are widely deployed for enterprise security. These monitors continuously record system-level activities as audit logs, capturing many aspects of the system's execution states. When reacting to a security incident, cyber analysts perform a causality analysis on audit logs to discover the root cause of the attack and the scope of its damages [3]. However, the amount of audit logs generated by a normal system is non-trivial. To overcome this challenge, recent research solutions scale up causality analysis by eliminating irrelevant [4]. Unfortunately, these solutions do not capture the semantics behind audit data and leave behavior recognition to analysts. Consequently, intensive manual effort is still involved in evaluating relevant yet benign and complicated events that dominate audit logs.

The main problem faced by analysts is a *semantic gap* between low-level audit events and high-level system behaviors. Existing work strives to bridge this gap by matching audit events against a knowledge store of expert-defined rules that describe behaviors, such as tag-based policies [2], query graphs [5], and TTP specifications [1]. Essentially, these solutions identify high-level behaviors through tag propagation or graph matching. However, an expected bottleneck is the manual involvement of domain experts to specify such rules. For example, MORSE [2] needs experts to traverse system entities (e.g., files) and initialize their confidentiality and integrity tags for tag propagation. TGMiner [5] requires manual behavior labeling in training log sets before mining discriminative behavioral patterns and searching for their existence in test sets. Despite crucial role in audit log analysis, mapping events to behaviors heavily relies on expert knowledge, which may hinder its applications in practice.

Extracting representative behaviors from audit events for analyst investigation provides an efficient strategy to mitigate this problem. More concretely, we can use procedural analysis to automatically abstract high-level behaviors and cluster semantically similar ones, albeit without the labels explaining what they are. However, because repetitive/comparable behaviors have already been clustered, analysts only need to label the representatives from clusters, resulting in far fewer events to be investigated. Besides reducing the manual workload in behavior analysis, automatic behavior abstraction also enables proactive analysis to detect unusual behavioral patterns in insider threats or external exploits. Particularly, any deviation of normal behaviors can be flagged efficiently for attack response.

While behavior abstraction sounds promising, there are two main challenges to extracting behaviors and inferring their semantics: event semantics differentiation and behavior identification. Audit events record general-purpose system activities and thus lack knowledge of high-level semantics. A single event, such as process creation or file deletion, can represent different semantics in different scenarios. Furthermore, due to the large-scale and highly interleaving nature of audit events, partitioning events and identifying boundaries of behaviors are like finding needles in a haystack.

To address the above challenges, our first key insight is that the semantics of audit events can be revealed from the contexts in which they are used. Intuitively, we can represent behaviors by aggregating the semantics of their constituent events. With such representations, similar behaviors can be clustered together. In addition, we observe that the information

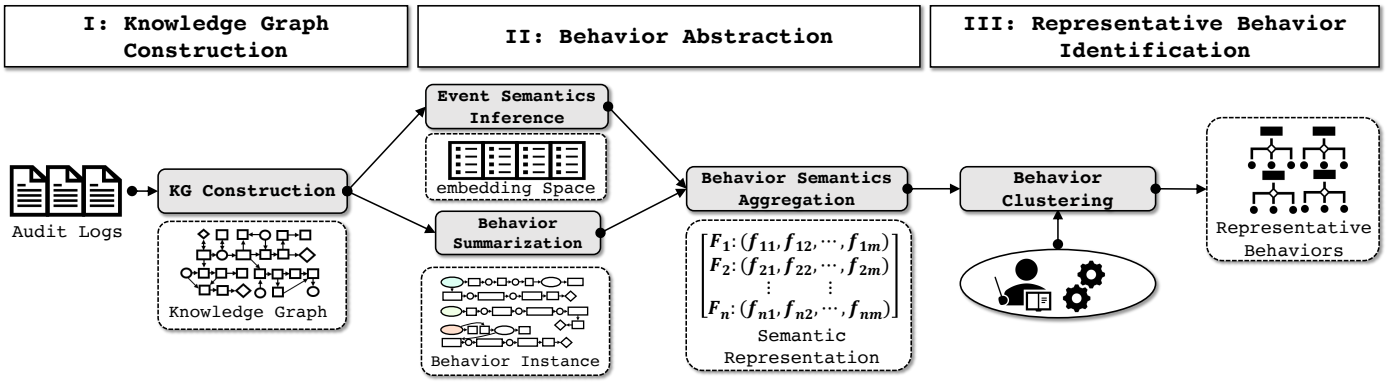


Fig. 1: WATSON Overview.

flow of system entities provides a natural boundary of high-level behaviors. It can serve as guidance to correlate audit events belonging to individual behaviors.

In this paper, we present WATSON, an automated behavior abstraction approach that aggregates the semantics of audit events to model behavioral patterns. It does not assume expert knowledge of event semantics to perform behavior abstraction. The semantics is obtained automatically from the context of event usage in audit logs. We call this the *contextual semantics* of events. More specifically, WATSON first leverages a translation-based embedding model to infer the semantics of audit events based on contextual information in logs. Then, WATSON identifies events connected to related data objects (i.e., files and network sockets) and aggregates their semantics as the representation of high-level behaviors. Finally, WATSON clusters similar behaviors recorded in audit logs and distinguishes the representatives for analyst investigation.

II. WATSON OVERVIEW

Figure 1 shows the overview of WATSON. It consists of three phases: Knowledge Graph Construction, Behavior Abstraction, and Representative Behavior Identification. WATSON takes as inputs system audit data. It supports events from Linux and Windows systems. It summarizes behavior instances, uncovers their semantics, and finally outputs representative high-level behaviors. Specifically, given audit logs in a user session as the input, the Knowledge Graph Construction module first parses logs into triples and constructs the log-based knowledge graph (KG). Then, the Event Semantics Inference module employs a translation-based embedding model to infer the contextual semantics of nodes in the KG. At the same time, the Behavior Summarization module enumerates subgraphs from the KG to summarize behavior instances. Combined with node semantics, the Behavior Semantics Aggregation module next enhances subgraphs to encode the semantics of behavior instances. Finally, the Behavior Clustering module groups semantically similar subgraphs into clusters, each specifying a high-level behavior. These cluster-based behavior abstractions can further be used to reduce the efforts of downstream tasks. We built an end-to-end system to facilitate audit analysis.

III. EVALUATION

We evaluate WATSON's correctness and explicability using 17 daily routines and eight real-life attacks simulated in an enterprise environment. In addition, we use the public DARPA TRACE dataset [1] released by the DARPA Transparent Computing program to evaluate WATSON's efficacy in attack investigation. We note that WATSON is the first to abstract both benign and malicious behaviors for evaluation. Previous techniques do not take benign behaviors into consideration because they mainly focus on attack detection. However, WATSON extracts high-level behaviors regardless of their security concerns. Experimental results show that WATSON accurately correlates system entities with similar usage contexts and achieves an average F1 score of 92.8% in behavior abstraction. Moreover, WATSON is able to reduce the number of audit logs for analyst investigation by two orders of magnitude.

ACKNOWLEDGEMENT

This research is supported by the National Research Foundation, Singapore under its Industry Alignment Fund – Pre-positioning (IAF-PP) Funding Initiative, National Research Foundation, Singapore under its International Research Centres in Singapore Funding Initiative and Ministry of Education Humanities and Social Science project, China (No.16YJC790123), Beijing Natural Science Foundation (No. 4202036), and National Natural Science Foundation of China (No. 61871023). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

REFERENCES

[1] Wajih Ul Hassan, Adam Bates, and Daniel Marino. Tactical provenance analysis for endpoint detection and response systems. In *IEEE Security and Privacy*, 2020.

[2] Md Nahid Hossain, Sanaz Sheikhi, and R Sekar. Combating dependence explosion in forensic analysis using alternative tag propagation semantics. In *IEEE Security and Privacy*, 2020.

[3] Samuel T King and Peter M Chen. Backtracking intrusions. In *SOSP*, 2003.

[4] Shiqing Ma, Juan Zhai, Yonghwi Kwon, Kyu Hyung Lee, Xiangyu Zhang, Gabriela Ciocarlie, Ashish Gehani, Vinod Yegneswaran, Dongyan Xu, and Somesh Jha. Kernel-supported cost-effective audit logging for causality tracking. In *USENIX ATC*, 2018.

[5] Bo Zong, Xusheng Xiao, Zhichun Li, Zhenyu Wu, Zhiyun Qian, Xifeng Yan, Ambuj K Singh, and Guofei Jiang. Behavior query discovery in system-generated temporal graphs. In *VLDB*, 2015.

[1] https://github.com/darpa-i2o/Transparent-Computing/blob/master/README-E3.md

Poster: Watson: Abstracting Behaviors from Audit Logs via Aggregation of Contextual Semantics

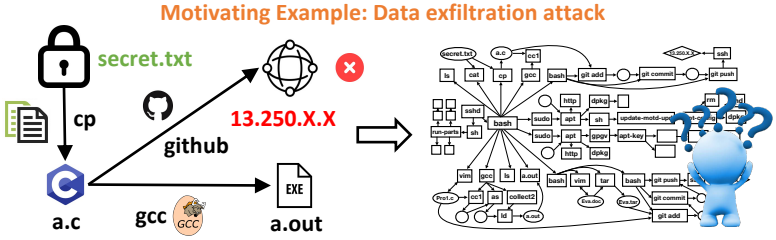
Jun Zeng, Zheng Leong Chua, Yinfang Chen, Kaihang Ji, Zhenkai Liang, Jian Mao
{junzeng,yinfang,kaihang,liangzk}@comp.nus.edu.sg,
czl@iiyume.org, maojian@buaa.edu.cn

Background and Motivation

Endpoint monitoring solutions record audit logs for attack investigation. **Audit logs** are a history of events representing OS-level activities and provide deep visibility into security incidents with data provenance.

Researchers propose to use a provenance graph to navigate through audit logs, but real-world audit logs are always **large-scale** and provenance graphs can easily **overwhelm** security analysts.

Motivating Example: Data exfiltration attack



Related works:

- Scale up provenance analysis:
  - Data reduction [NDSS'16, 18 …] & Query system [Security'18, ATC'18 …]
  - Recognizing behaviors of interest requires intensive manual efforts

A semantic gap between low-level events and high-level behaviors

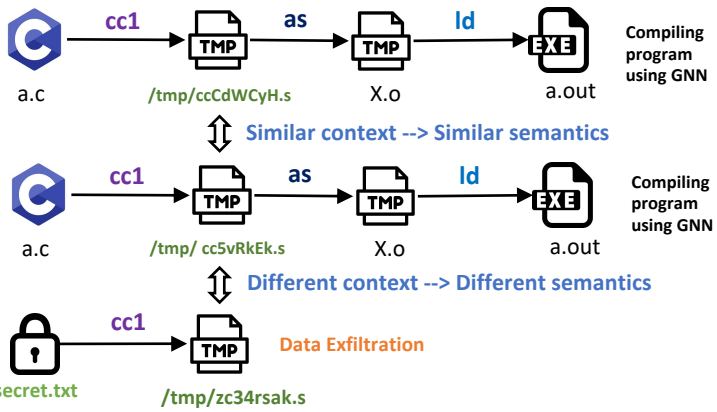
- Apply expert-defined specifications to bridge the gap
  - Match audit events against domain rules that describe behaviors
  - Query graph [VLDB'15, CCS'19], Tactics Techniques Procedures (TTPs) specification [SP'19,20], and Tag policy [Security'17,18]

Behavior-specific rules heavily rely on domain knowledge (**time-consuming**)

How can we automatically **abstract** high-level behaviors from low-level audit logs and **cluster** similar behaviors to assist investigation?

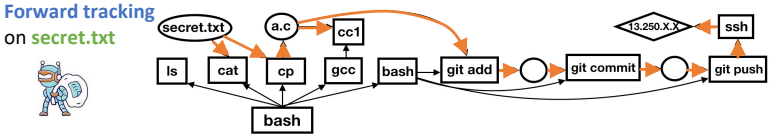
Our insight

- How do analysts manually interpret the semantics of audit events?



The semantics of audit events can be revealed from their usage **contexts** in logs

- How do analysts manually identify behaviors from audit events?

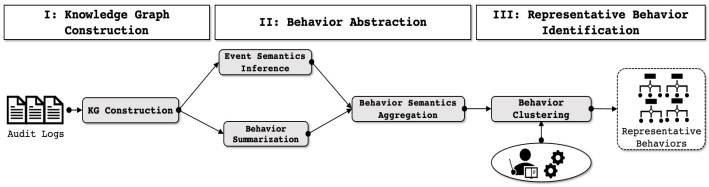


Behaviors can be summarized by tracking information flows rooted at data objects

System Design: Watson

An automated behavior abstraction approach that **aggregates the semantics of audit logs to model behavioral patterns**

- Input: audit logs (e.g., Linux Audit[1])
- Output: representative behaviors

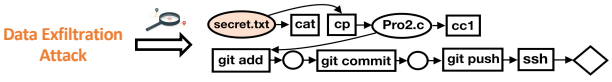


KG Construction: We propose to use a knowledge graph (KG), a directed acyclic graph built upon triples, to represent audit logs:

$$\mathcal{KG} = \{(h, r, t) | h, t \in \{Process, File, Socket\}, r \in \{Syscall\}\}$$

Behavior Abstraction: (1) We use knowledge-graph embedding (i.e., TransE) to capture contextual semantics of audit logs; **TransE: Head + Relation ≈ Tail → Context decides semantics**

(2) We identify individual behaviors by applying an adapted depth-first search to track information flows rooted at a data object



Representative Behavior Identification: (1) We aggregate the event semantics to represent behaviors semantics (noise reduction + attention: IDF);
(2) We cluster semantically similar behaviors using agglomerative hierarchical clustering analysis (HCA) and extract the most representative behaviors for security analysis for attack investigation

Evaluation Results

- **Experimental Setup:**
  - Simulated dataset: **275,863,292** events in 4,280 SSH sessions
  - DARPA Trace Dataset[2]: **726,072,596** events in 211 graphs

- **Behavior Abstraction Accuracy:**

Can WATSON cluster similar behaviors?

High F1 score on both benign and malicious behavior abstraction

- **Event Semantics Explicability:**

Does inferred event semantics match our domain knowledge?

Semantically similar system entities are **clustered** in the embedding space

- **Efficacy in Attack Investigation:**

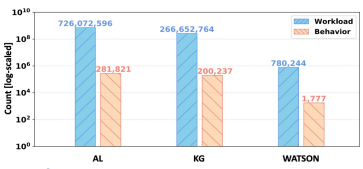
How many manual efforts can WATSON save?

Two orders of magnitude reduction in analysis workload and behaviors

- **Accuracy on 17 daily routines and 8 attacks in the simulated dataset**

Behavior	Recall	Precision	F1
Package Installation	95.3%	97.9%	96.6%
Data Theft	100%	100%	100%
…	…	…	…
Average	94.2%	92.8%	92.8%

- **Analysis workload reduction in the analysis of APT attacks in the DARPA Trace dataset**



- **Visualization (t-SNE) on the embedding space of 25 data object and 53 programs**