Poster: LogCraft: Crafting CVE-Aware Synthetic Worlds (Logs)

Kai-Xian Wong*
University of Malaya
Kuala Lumpur, Malaysia
wong020605@gmail.com

Chan-Jien Tan*
University of Malaya
Kuala Lumpur, Malaysia
tw.tanchanjien@gmail.com

Yi-Ting Huang
National Taiwan University of Technology and Science
Taipei, Taiwan
ythuang@mail.ntust.edu.tw

Ying-Ren Guo
Academia Sinica
Taipei, Taiwan
hitoshi@iis.sinica.edu.tw

Yu-Zih Jheng
National Taiwan University
Taipei, Taiwan
r13725036@ntu.edu.tw

Guo-Wei Wong
National Taiwan University
Taipei, Taiwan
d10922026@ntu.edu.tw

Meng Chang Chen
Academia Sinica
Taipei, Taiwan
mcc@iis.sinica.edu.tw

Abstract

The generation of realistic, labeled audit logs is critical for evaluating the effectiveness of cybersecurity detection systems, yet existing datasets often lack diversity, scalability, and support for real-world vulnerabilities. While SAGA provides a solid foundation by addressing many of these challenges, this work further extends its capabilities in two key directions. First, we expand the coverage of attack behaviors by simulating additional lateral movement techniques and introducing a semi-automated process for generating labeled log templates. Second, we enable CVE-based log synthesis by incorporating vulnerability information into the attack modeling process. These enhancements improve the fidelity, automation, and extensibility of synthetic log generation, supporting more effective training and evaluation of threat detection models.

CCS Concepts

• **Security and privacy → Malware and its mitigation**; **Intrusion detection systems**.

Keywords

Synthetic Audit Log, Large Language Models, APT campaign

ACM Reference Format:
Kai-Xian Wong, Chan-Jien Tan, Yi-Ting Huang, Ying-Ren Guo, Yu-Zih Jheng, Guo-Wei Wong, and Meng Chang Chen. 2025. Poster: LogCraft: Crafting CVE-Aware Synthetic Worlds (Logs). In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security (CCS '25), Oct. 13–17, 2025, Taipei, Taiwan.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3719027.3760736

*Both authors contributed equally to this research.

CCS '25, Taipei, Taiwan

ACM ISBN 979-8-4007-1525-9/2025/10
https://doi.org/10.1145/3719027.3760736

1 Introduction

The emergence of sophisticated threats such as Advanced Persistent Threats (APTs) brings greater challenges to the cybersecurity community, such as BlackEnergy [4] and SolarWinds Compromise [5]. These multi-stage attack campaigns typically involve initial compromise, lateral movement, privilege escalation, and data exfiltration. While detection and defense techniques have continued to evolve, their development and validation often rely on high-quality, labeled audit logs to simulate realistic attack scenarios and assess system responses. Unfortunately, the availability of such data is limited, with public datasets either too narrow in scope, insufficiently labeled, or withheld due to privacy concerns.

Recent advances such as the SAGA [6] framework have demonstrated the value of synthetic audit log generation for simulating APT campaigns in a structured and reproducible manner. By leveraging red-team traces and aligning with the MITRE ATT&CK framework, SAGA provides fine-grained, labeled logs that support both rule-based and machine learning–based detection research.

Building upon this foundation, we extend SAGA with two key enhancements. First, we simulate a wider range of lateral movement techniques in a realistic multi-host Active Directory environment. Second, we introduce CVE-driven log generation by using LLMs to map vulnerabilities to ATT&CK techniques and extract behavioral traces from PoC exploits. These upgrades improve the realism, automation, and scalability of SAGA, enabling better evaluation of threat detection models.

2 Methodology

Building on our previous SAGA framework, this work retains its core design while expanding attack coverage, automating template generation, and integrating CVE-based exploit simulation. Figure 1 illustrates the overall architecture of our extended SAGA-based pipeline.

2.1 SAGA

SAGA is a framework designed to generate realistic, labeled audit logs that simulate advanced persistent threat (APT) behaviors. It

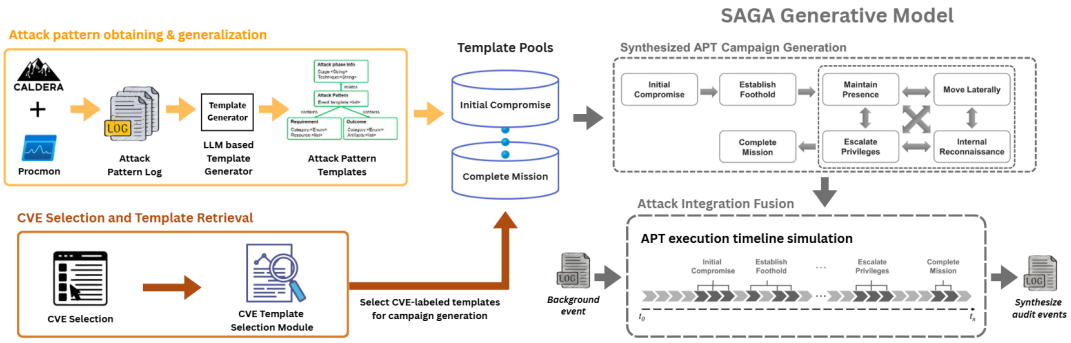

Figure 1: Overview of the enhanced SAGA pipeline, highlighting key improvements and extensions.

addresses the lack of large-scale, high-fidelity audit data necessary for evaluating and training cybersecurity analytics.

SAGA generates synthetic logs by combining benign activities with red-team attack traces labeled with MITRE ATT&CK techniques. Its architecture supports high-fidelity log simulation, offering fine-grained technique annotations and abstracted campaign-level views. These features enable both rule-based and machine learning approaches to evaluate detection strategies using synthetic yet realistic data.

Building on the core concepts of SAGA, we propose a series of functional enhancements to extend its applicability, such as expanding attack scenarios and introducing CVE-driven attack synthesis.

2.2 Attack Coverage Expansion and Template Automation

Lateral movement has become a critical step in many modern attack campaigns, allowing threat actors to move between systems and escalate privileges within a network. To reflect its growing importance in real-world threats, we expanded SAGA's coverage by simulating a broader range of lateral movement techniques—including PsExec, WinRM, VNC, and SMB-based exploits like EternalBlue.

To simulate these techniques in a controlled environment, we used Caldera [1] as the attack execution platform and Process Monitor [7] to capture system-level activity. The simulations were carried out in a custom Active Directory setup mimicking a corporate network, consisting of one Windows Server 2022 domain controller, and two Windows 10 Pro clients. Each simulation was repeated under identical baseline conditions (domain controller, two client machines, identical network policies), with only the attack vector varied (PsExec, WinRM, VNC, SMB/EternalBlue). This ensured comparability across scenarios while maintaining controlled variability in the lateral movement technique. This setup enables us to simulate and capture realistic attack traces for generating SAGA templates.

However, while the simulation and logging process is relatively straightforward, manually creating SAGA templates from raw logs is not efficient. A single simulation can generate hundreds of thousands to millions of events, and filtering these to extract relevant behavior for each technique is time-consuming and prone to inconsistency. To overcome this bottleneck, we introduced a semi-automated template generation pipeline using large language models (LLMs). Here, we use Gemini in this study.

The template generation pipeline consists of four stages. First, we preprocess the raw logs by applying basic filters to retain only relevant event types based on operations of interest. Next, an LLM is used to generate event filtering rules. The input to the LLM includes the attack commands and a simplified version of the filtered logs (process name, operation type, path, and details) to avoid exceeding token limits. These rules are then applied to the logs to extract relevant events to the simulated behavior. In the third stage, the extracted events and original attack command are sent to the LLM again to generate semantic annotations (e.g., requirements, hypernyms, and outcomes). Finally, all outputs are automatically structured into the SAGA template format using predefined schemas. The resulting templates can also be manually reviewed or modified if needed to refine accuracy or add contextual details.

This process notably reduces manual effort, improves consistency, and allows us to scale the template creation process across various techniques. It also lays the groundwork for broader, tactic-wide automation in the future.

2.3 CVE-Based Audit Log Generation

To simulate realistic attack traces based on real-world vulnerabilities, we proposed a CVE-based audit log generation capability. This feature allows users to select a specific Common Vulnerabilities and Exposures (CVE) identifier and automatically generate an audit log that contains behaviors associated with the vulnerability.

Our approach begins by mapping CVEs to MITRE ATT&CK techniques. We adopted the mapping methodology from the MITRE Center for Threat-Informed Defense's Mapping Explorer [2], which categorizes the exploitation process into three components:

- **Exploitation Technique:** The method used to exploit the vulnerability.
- **Primary Impact:** The initial benefit gained through the exploitation of the vulnerability.
- **Secondary Impact:** The action that the adversary can take after gaining the benefit of the primary impact.

To address the limited number of CVEs in the Mapping Explorer dataset, we developed a Large Language Model (LLM) agent (Gemini) to generate mappings. The agent receives structured CVE descriptions from authoritative sources like the CVE Program's *cvelist* [3] on GitHub. Through carefully designed prompts, the LLM agent is guided to analyze these descriptions and apply our mapping methodology. This approach allows us to generate consistent and accurate ATT&CK technique mappings without fine-tuning the model.

To capture the actual system-level behavior of a CVE, we execute its proof-of-concept (PoC) exploit in a secure and isolated test environment. During exploitation, system-level activities are recorded using Process Monitor for behavioral analysis.

Following both the technique mapping and behavioral data collection, we identify and label relevant SAGA abilities that closely match the observed activity. These labeled abilities are then directly utilized by the system to generate an audit log when a user selects a CVE. Each ability is integrated into its corresponding stage of the attack lifecycle, and any remaining stages are populated using our standard audit log generation process. This approach enables the generation of comprehensive and high-fidelity audit logs that incorporate CVE exploitation within the overall attack lifecycle.

3 Case Study

In this section, we demonstrate the capabilities of our system using a case study focused on CVE-2021-34527, a vulnerability in the Windows Print Spooler that allows remote attackers to execute arbitrary code with SYSTEM-level privileges. Each CVE undergoes a preprocessing step, where it is first mapped to relevant MITRE ATT&CK techniques using the methodology described in Section 2.3. Next, a proof-of-concept (PoC) exploit is executed in a controlled environment to capture system-level behavior using Process Monitor.

The recorded behavior is then compared with existing SAGA templates to identify templates that contain matching events under the same technique. Matching templates are labeled with the CVE and stored for later use. For example, the CVE-2021-34527 exploit generated events such as 'DLL creation', 'service start', 'account creation', 'registry modification', and 'account creation'. These events were mapped to a template corresponding to the move laterally, escalate privilege, and maintain presence stage.

When a user selects CVE-2021-34527 in the system, SAGA retrieves the labeled templates and inserts them into the corresponding attack lifecycle stages, as shown in Figure 2. Stages not affected by the CVE are populated with other templates to ensure a coherent APT campaign. This approach allows the system to generate realistic CVE-driven audit logs while maintaining the structure and fidelity of SAGA templates.

The generated audit logs preserve the attack sequence and provide realistic system-level behavior, making them suitable as a dataset for evaluating detection mechanisms or for training machine learning models in cybersecurity research.

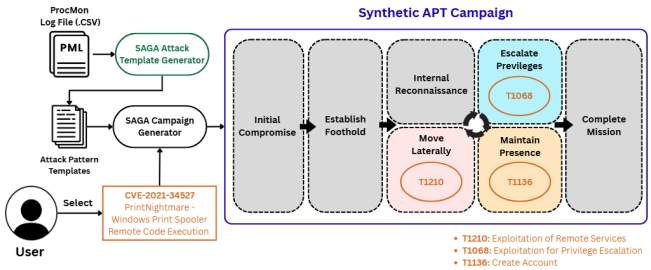

Figure 2: Illustration of CVE-driven template selection and log generation.

4 Conclusion and Future Work

In this work, we enhanced the SAGA system with two extensions: automated template generation for expanded coverage and CVE-based audit log creation. We simulated additional lateral movement techniques such as PsExec, WinRM, VNC, and SMB exploits in a multi-host Active Directory environment, converting the logs into structured templates via a semi-automated, LLM-assisted pipeline that reduced manual effort and improved consistency. Additionally, by mapping CVEs to ATT&CK techniques using structured data and LLM inference — and validating through PoC execution — we identified and tagged behaviorally aligned templates, enabling users to generate realistic, CVE-driven synthetic attack campaigns for evaluating detection systems.

In the future, we will focus on two key areas. First, we plan to implement a continuous integration pipeline that leverages our LLM agent to automatically map and incorporate new CVEs as they emerge, providing a constantly updated threat landscape for simulation. Second, we aim to enhance the LLM-assisted template generation pipeline by improving event filtering and exploring multi-agent collaboration to further automate the process and reduce manual effort.

5 Acknowledgments

This work was supported by NSTC, R.O.C. under Grant No. 114-2221-E-001-018, 114-2221-E-011-080-MY3, and 113-2634-F-001-002-MBK.

References

[1] Andy Applebaum, Doug Miller, Blake Strom, Chris Korban, and Ross Wolf. 2016. Intelligent, automated red team emulation. In *Proceedings of the 32nd Annual Conference on Computer Security Applications*.
[2] Center for Threat-Informed Defense. 2024. Mapping CVEs to ATT&CK Techniques. https://center-for-threat-informed-defense.github.io/mappings-explorer/about/.
[3] CVE Program. 2024. cvelist GitHub Repository. https://github.com/CVEProject/cvelistV5.
[4] E-ISAC. 2016. Analysis of the Cyber Attack on the Ukrainian Power Grid. https://nsarchive.gwu.edu/sites/default/files/documents/3891751/SANS-and-Electricity-Information-Sharing-and.pdf.
[5] FireEye. 2020. Highly Evasive Attacker Leverages SolarWinds Supply Chain to Compromise Multiple Global Victims With SUNBURST Backdoor. https://www.mandiant.com/resources/blog/evasive-attacker-leverages-solarwinds-supply-chain-compromises-with-sunburst-backdoor.
[6] Yi-Ting Huang, Ying-Ren Guo, Yu-Sheng Yang, Guo-Wei Wong, Yu-Zih Jheng, Yeali Sun, Jessemyn Modini, Timothy Lynar, and Meng Chang Chen. 2024. SAGA: Synthetic Audit Log Generation for APT Campaigns. *arXiv preprint arXiv:2411.13138* (2024).
[7] Microsoft Sysinternals. 2023. Process Monitor v3.96. https://learn.microsoft.com/en-us/sysinternals/downloads/procmon.