From Observations to Insights: Constructing Effective Cyberattack Provenance With PROVCON

Anis Yusof*, Shaofei Li†, Arshdeep Singh Kawatra*, Ding Li†, Ee-Chien Chang* and Zhenkai Liang*
*National University of Singapore
†Key Laboratory of High-Confidence Software Technologies (MOE), School of Computer Science, Peking University
*{anis, changec, liangzk}@comp.nus.edu.sg, *arshdeepsk@u.nus.edu, †{lishaofei, ding_li}@pku.edu.cn

***Abstract*—To improve the preparedness of Security Operation Center (SOC), analysts may leverage provenance graphs to deepen their understanding of existing cyberattacks. However, the unknown nature of a cyberattack may result in a provenance graph with incomplete details, thus limiting the comprehensive knowledge of the cyberattack due to partial indicators. Furthermore, using outdated provenance graphs imposes a limit on the understanding of cyberattack trends. This negatively impacts SOC operations that are responsible for detecting and responding to threats and incidents. This paper introduces PROVCON, a framework that constructs a provenance graph representative of a cyberattack. Based on documented cyberattacks, the framework reproduces the cyberattack and generates the corresponding data for attack analysis. The knowledge gained from existing cyberattacks through the constructed provenance graph is instrumental in enhancing the understanding and improving decision-making in SOC. With the use of PROVCON, SOC can improve its cybersecurity posture by aligning its operations based on insights derived from documented observations.**

I. INTRODUCTION

The critical operations in SOC require analysts to continuously improve their understanding of cyberattack trends. To get a clearer understanding of existing cyberattacks, analysts may rely on provenance graphs to discover how an attack unfolded, determine the affected systems, identify the compromised data, and eventually pinpoint the incident's root cause. This enhances their understanding and ability to explain correlated malicious activities that occurred during the cyberattack [57], especially during threat hunting [31], [39] and event reconstruction [11]. The reliance on the insights derived from provenance graphs makes them vital in delivering a positive impact towards SOC operations, such as training on realistic attack scenarios, making informed decisions, and refining Digital Forensics and Incident Response (DFIR) techniques.

Acquiring a comprehensive provenance graph that closely reflects contemporary cyberattacks is pivotal in the context of the SOC [18]. Relying on provenance graphs with incomplete and outdated information will undermine its effectiveness in providing valuable insights for SOC operations. One way to address this challenge is to rely on existing cyberattack datasets (e.g., logs, traces) to derive provenance graphs. However, obtaining such datasets is challenging due to the scarcity of real-world cyberattack data that may not be widely shared due to privacy concerns. Despite utilizing the publicly available cyberattack datasets, the provenance graph may be incomplete as it does not contain comprehensive indicators about the cyberattack. Furthermore, the insights drawn from the provenance graph become obsolete as they are only relevant for the period during which the cyberattack occurred (e.g., attack techniques, software configurations). Using an incomplete and outdated provenance graph may lead to a lack of understanding about emerging and evolving cyberattack activities. This raises the need for a provenance graph that is representative of current cyberattacks.

To address the incomplete and outdated datasets that provide limited value for attack understanding, we propose PROVCON, a framework for constructing provenance graphs. We first extract the latest cyberattack information from Cyber Threat Intelligence (CTI) reports as primitives and collect them as a specification. Based on the specification, we deploy the cyberattack within a cyber range and execute the attack activities before extracting the data that consists of attack indicators. The attack data is then transformed into a provenance graph, which contains insights about the cyberattack. We highlight the need to produce provenance graphs that contain comprehensive attack information, as having this capability augments the understanding about existing cyberattacks for SOC analysts. In summary, our contributions are as follows:

- We propose a framework called PROVCON for constructing provenance graphs that augment the understanding of existing cyberattacks in SOC.
- We conduct a case study to demonstrate the usefulness of PROVCON in constructing complete and timely provenance graphs that enhance the understanding of existing cyberattacks.
- We publicly share the repository[1] that contains the provenance graph and corresponding data constructed using PROVCON.

Workshop on Security Operation Center Operations and Construction (WOSOC) 2025
24 February 2025, San Diego, CA, USA
ISBN 979-8-9919276-0-4
https://dx.doi.org/10.14722/wosoc.2025.23008
www.ndss-symposium.org

TABLE I: Summary of existing public datasets.

Dataset	Release Date	Collector	Papers that use the dataset	Heterogeneous Data	Heterogeneous Hosts	Up-to-date Attacks	Completeness of Indicators	Event-level Ground Truth
StreamSpot [34]	2016	SystemTap [28]	[13], [23]–[25], [29], [52], [54]	✘	✘	✘	✘	✘
DARPA E3 [1]	2018	Auditd [46], DTrace [4], ETW [17]	[8], [13], [18], [19], [23]–[27], [29], [32], [39], [41], [44], [52]–[56]	✘	✔	✘	✘	✘
DARPA E5 [3]	2020	Auditd, DTrace, ETW	[13], [15], [27], [44]	✘	✔	✘	✘	✘
DARPA OpTC [2]	2020	Auditd, DTrace, ETW	[52]	✘	✔	✘	✘	✘
UNICORN [25]	2020	CamFlow [45]	[25], [29], [52]	✘	✘	✘	✘	✘
ATLAS [7]	2021	Windows Security Auditing [38], Mozilla Firefox [43]	[7], [16], [24], [48], [54]	✔	✘	✘	✔	✔
NodLink [32]	2024	Sysdig [50], ETW	[32]	✘	✔	✘	✔	✔
TREC [40]	2024	Kellect [12]	[40]	✘	✘	✘	✔	✘

II. Background

To construct provenance graphs that are relevant for SOC operations, we need to identify the features that make a provenance graph effective in capturing comprehensive and up-to-date indicators. Understanding existing datasets and gathering requirements is key to constructing provenance graphs applicable to attack analysis. This section provides a comprehensive study of public datasets used by provenance analysis papers published at the top conferences during 2019-2024, including IEEE S&P, USENIX Security, CCS, and NDSS. We carefully read these papers and summarized the public datasets that they used for their evaluations. Finally, we found eight public datasets that are widely used by the research community. We summarize these datasets and analyze them from five perspectives: heterogeneity of data, heterogeneity of hosts, up-to-date attacks, completeness of indicators, and event-level ground truth. The results are shown in Table I.

Heterogeneous Data. Existing datasets are mainly provenance audit data from each host respectively. However, existing attacks often exploit multiple hosts in a network and laterally move to the target to achieve their malicious purposes [9]. Therefore, network traffic is also important to capture the whole attack campaign. In addition, different detection methods may require different types of data. For example, attackers often use stealthy techniques to manipulate the memory of compromised processes. So, the memory dump of the compromised hosts is also useful for detecting malicious activities. Therefore, provenance data from separate hosts is insufficient to capture and analyze real-world attack campaigns as a whole. In our study, only ATLAS [7] provides heterogeneous data containing DNS logs and Web browser traffic. So, we observe the need to provide heterogeneous data for provenance analysis research.

Heterogeneous Hosts. In real-world scenarios, there are different operating systems in SOC, such as Windows, Ubuntu, FreeBSD and MacOS. The attackers will use different attack techniques or tools based on the host's operating systems to achieve their malicious purposes. For example, attackers often use Cobalt Strike [22] to control Windows hosts and Metasploit [37] to penetrate Linux hosts. Hence, it is important to provide provenance data from different operating systems to evaluate the generalizability of the proposed methods. However, only half of the datasets provide provenance data from different operating systems.

Up-to-date Attacks. One characteristic of Advanced Persistent Threats (APT) attacks is that the attackers will use zero-day vulnerabilities or the latest attack techniques to evade detection from security tools used by the victims. Therefore, the datasets should contain the latest attack techniques for evaluation. However, all of the datasets use out-of-date attacks, such as CVEs that are released four years before the dataset release date [7] and the scripts from the Atomic Red Team [32], [40], [51]. However, utilizing the latest attacks for evaluation requires a lot of manual effort with respect to setting up testbeds and collecting data. Thus, an automated way to generate the datasets with the latest attacks will be helpful.

Completeness of Indicators. In SOC, the detection and investigation of APT attacks often require complete indicators of these attacks, such as the command line of the malicious processes, malicious files, and malicious IP addresses. However, some commonly used public datasets do not contain complete indicators of the attacks. For instance, the StreamSpot [34] dataset retains only the graph structure of the provenance data, omitting specific details of the nodes and edges. This incompleteness complicates the evaluation using methods that rely on complete attack indicators. DARPA E3 [1] and DARPA E5 [3] provide a list of indicators in their documents, but some of these indicators cannot be found in the provenance data. Therefore, it is necessary to provide complete indicators of cyberattacks in the datasets to support all types of provenance analysis methods.

Event-level Ground Truth. The ground truth of datasets is important for the evaluation of the detection system in SOC. Some datasets, such as StreamSpot [34] and Unicorn [25] only label the anomaly in graph-level granularity, which makes it difficult to evaluate the event-level accuracy. The DARPA E3 [1] and DARPA E5 [3] provide documents that describe the attack scenarios and the indicators of the attacks, but it takes a lot of manual effort to label the ground truth in the datasets. In SOC, analysts often need to investigate malicious activities in event-level granularity to understand the attack campaigns and take action [18]. Event-level ground truth is necessary for

[1]https://github.com/NUS-Curiosity/provcon

the evaluation of Provenance-Based Endpoint Detection and Response (P-EDR).

Limitations. Based on these observations, we find that the existing public datasets have limitations at least in two of the five perspectives. For example, DARPA E3 [1] and StreamSpot [3] are the most widely used datasets, but they cannot simulate the APT attacks well and require a lot of manual effort to evaluate them. Some researchers have claimed that DARPA E3 is not suitable for evaluating their proposed methods [7], [40], [44] thus they need to collect and build their own datasets. However, most of the simulated datasets are not publicly available, which makes it difficult for other researchers to reproduce the results. In addition, Deep Nerual Network (DNN), Graph Neural Network (GNN), and Large Language Model (LLM) are widely used by existing P-EDR. The data quality is crucial for the performance of these models. Therefore, improving the quality of public datasets is necessary for the research community to develop and evaluate the P-EDR.

Requirements. To address these limitations, we summarize the requirements for public datasets used in provenance analysis research. First, the datasets should contain heterogeneous data so that it is supported for different downstream tasks. Second, the datasets should contain provenance data from different operating systems and mimic the latest attacks to evaluate their effectiveness and generalizability. Third, the datasets should contain complete indicators of attacks and event-level ground truth for model training and the calculation of performance metrics. These requirements ensure that the provenance graph generates high-quality insights that are useful for attack investigation.

To satisfy these requirements, we propose a framework to construct provenance graphs from the latest CTI reports. The framework extracts the necessary attack information from the latest CTI reports and provisions a cyber range on a testbed to simulate the attack before collecting the heterogeneous data. The details of PROVCON are described in the next section.

III. METHODOLOGY

To generate a provenance graph that contains complete indicators and acts as a representation of current cyberattacks, we propose a framework called PROVCON. The framework identifies relevant cyberattack information from CTI reports and constructs a provenance graph corresponding to the cyberattack information, as described in Figure 1. Specifically, our framework first analyzes CTI reports as the input to PROVCON. The analysis focuses on recognizing cyberattack primitives that are relevant for generating a provenance graph. The primitives are then transformed into a cyberattack description that formalizes the primitives into technical requirements. These technical requirements are then used as a specification to reproduce the cyberattack by deploying a cyber range. The cyber range replicates the components and executes the attack events in a controlled environment. At the end of the execution, relevant data are extracted from the respective hosts involved in the cyber range. This includes a variety of logs that contain key details found in the cyberattack. The logs are finally transformed into a provenance graph, which is used for attack investigation within SOC. To ease the investigation process, the extracted data are annotated based on the description used for cyberattack reproduction. By using PROVCON, the constructed provenance graph is enriched with indicators that are described in the CTI report while simultaneously having the data annotated based on the formalized cyberattack description.

A. Cyberattack Primitives Recognition

Selecting a CTI report that contains recent cyberattacks allows PROVCON to construct provenance graphs that are relevant to current trends. CTI reports contain descriptions about APT attacks and are used as input for PROVCON. The objective of this component is to parse the CTI report and identify relevant cyberattack information for constructing the cyber range. This component is segregated into two stages. The first stage is to identify the cyberattack primitives that are relevant for constructing a provenance graph. This includes primitives that describe the infrastructure, tools, malware, indicators, and attack patterns that are involved in the cyberattack. Certain primitives are directly related to the provenance graph. This includes the tools and malware that are being used during the attack, attack patterns that indicate the malicious activities taken by the threat actors, and possible indicators that are observed. There are also primitives that are indirectly related to the provenance graph but are necessary for provenance graph construction. This includes infrastructure information such as database server and Command and Control (C2) server. These indirect primitives are required to derive the information found in the provenance graph. In the second stage, the identified cyberattack primitives are categorized into two groups, namely *environment* and *events*. The *environment* category contains all the primitives that are used to build the cyber range (e.g., software, operating systems, hosts, network configuration). The *events* category contains all the primitives that are used to form the sequence of events (e.g., benign operations, attack sequences). This categorization forms the fundamentals of cyberattack reproduction.

The categorized primitives are used to compose the cyberattack description. The cyberattack description specifies the instances that are involved in the cyberattack (e.g., Windows clients, Linux-based database server). Additionally, the cyberattack description specifies the artifacts that are involved in every instance, such as tools and applications. These artifacts may include vulnerable components that can be exploited during the attack. Network configurations (e.g., IP addresses, subnet) are also defined in the cyberattack description in order to provide network connectivity between the instances. Aside from the environment-related description, the cyberattack description also specifies the events that take place throughout the cyber range. These events may include benign or malicious activities that occur in the environment. Therefore, the cyberattack description is designed to capture customizable and complex cyberattack scenarios which include heteroge-

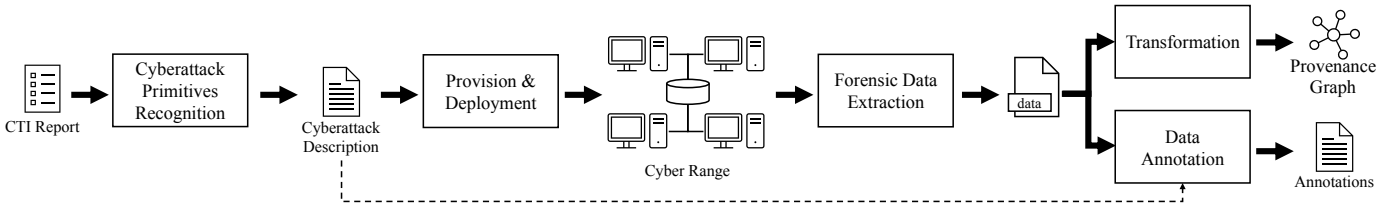

Fig. 1: Framework of PROVCON.

neous instances, applications, varying networking topology, and events.

B. Provision and Deployment

The cyberattack description is written in a structured and high-level custom Domain Specific Language (DSL), which provides an environment-level perspective of the cyber range written as code. Our custom language is designed to declaratively describe the environment and event information in a single language while being directly translatable to the underlying Infrastructure as Code (IaC). As an alternative, the cyberattack description can also be written directly as IaC for deployment. In PROVCON, we devise the cyberattack description using our custom DSL which is then translated to the corresponding underlying IaC (i.e., Vagrant and Ansible). We utilize Vagrant as a declarative and reproducible mechanism to realize the environment components in the cyber range. Additionally, Ansible is used as a configuration and orchestration tool to configure the deployed environment components and orchestrate the cyberattack activities inside the cyber range.

With the goal of realizing a cyber range in this component, the technical requirements defined in the cyberattack description are first translated to the respective IaC. The framework then iterates through each component and provisions the defined resources. This includes preparing the necessary applications, setting up the configurations, and retrieving virtual images for deployment. During deployment, the framework realizes the provisioned resources on our test bed. This includes the deployment of configured virtual instances and networking nodes along with their network configurations. Additionally, monitoring tools are configured and activated in every instance. This allows the monitoring of system-level and network-level activities, thus generating relevant logs that consist of indicators resulting from attack activities. As a result, this creates the initial state of the cyber range with all the necessary components for replicating the cyberattack.

Once the environment is ready, the framework iterates through the list of events and orchestrates the execution of activities throughout the cyber range. In every event, the artifacts (e.g., applications, tools, malware) are executed in the respective instance. Sequential execution of events ensures that the attack steps are controlled throughout the cyber range. Cyberattack reproduction is considered a success when all events have been executed successfully in the cyber range.

C. Forensic Data Extraction

After the cyberattack is reproduced, the cyber range contains data that is useful for analysis. The goal is to extract the necessary data, especially data that are relevant for constructing a provenance graph. In PROVCON, we extract system-level logs from the respective instances in the cyber range. The logs are generated by the monitoring tools that capture the activities that occur after the initial environment state. For Windows-based instances, we utilize the Windows Event Viewer, which captures both application and system messages. This includes application, system, and security logs that describe the events that occurred in the Windows-based instance. In addition to the default logging service in Windows, PROVCON uses System Monitor (Sysmon) as a supplementary monitoring tool to perform detailed logging of events that are not captured by the default logging service. Sysmon enhances the default logging capabilities in Windows Event Viewer by providing detailed visibility on processes (e.g., current and parent process ID), files (e.g., file creation timestamp), and network connections (e.g., IP address and port numbers). For Windows-based instances, the logs are extracted as Windows XML Event Log (EVTX) files. The EVTX format preserves the event information and its metadata, maintaining compatibility with various forensic analysis tools.

For Linux-based instances, we export all the generated logs located in the default log directory at `/var/log`. This directory includes various logs such as system logs, authentication logs, and application logs. In addition to the default logging in Linux, we utilize two additional monitoring tools, namely Linux Audit Daemon (auditd) and sysdig[2]. The goal of auditd is to monitor and record the responsible user and process for an event (e.g., identify user accessing a sensitive file), thus achieving audit-level capabilities. Additionally, sysdig focuses on capturing detailed system and application activities. The deep system-level visibility enables the identification and explanation of suspicious activities. To preserve the contents of the logs, all logs in `/var/log` are exported in their original format. This includes both auditd and sysdig logs (i.e., SCAP file).

In addition to system-level logs, we extract network-level logs from the respective instances. To monitor and capture network logs, we utilize Packet Monitor (Pktmon) for Windows-based instances and `tcpdump` for Linux-based instances. These network monitoring tools monitor and record the network communication from the instance's network interfaces,

[2]https://github.com/draios/sysdig

thus capturing network-level information during the course of events (e.g., source and destination IP addresses of packets). The network capture for every instance is extracted as a Packet Capture (PCAP) file, preserving the bytes observed on the network. Aside from the above logs, we also extract the memory of every instance by creating a memory dump. The dump contains various information about the instance, such as the process tree, loaded libraries, and active network connections. In this way, we preserve the state of the instance as an image that can be used by memory forensic tools (e.g., Volatility). By extracting heterogeneous types of data, PROVCON provides analysts with multi-perspective evidence that results from cyberattack reproduction.

D. Provenance Graph and Annotation

Data extracted from the compromised environment serves as a starting point for investigating a cyberattack and developing a comprehension of it. While analyzing the evidence in its original form, such data and indicators may not portray the intrinsic characteristics of a complex cyberattack. Therefore, a provenance graph is primarily used to provide information about the entire system execution. Hence, the objective of this component is to construct a provenance graph that satisfies the requirements identified in the earlier sections. Based on the heterogeneous data extracted from the cyber range, PROVCON utilizes the system-level logs as the foundation for constructing a provenance graph. The contents of the logs are parsed to identify causal relationships between entities such as users and resources. Mainly, PROVCON uses the deep visibility of system-level activities from sysdig and Sysmon to construct a provenance graph.

To investigate the cyberattack, analysts have to inspect and sieve the data to identify attack indicators. However, the constructed provenance graph and extracted cyberattack data are massive in terms of content. Identifying the indicators manually from this data is a laborious task. Furthermore, indicators that have been identified will need to be arranged in order to characterize the attack activities. This raises the need for automated annotation of data, enriching them with metadata that eases the investigation process for analysts. In PROVCON, annotation is performed on the extracted data. The annotation consists of attribution, tagging, and labels for the data. The annotation involves adding supplementary comments while attributing the responsible instances and artifacts that generate this data. Additionally, the data are tagged to the corresponding event. This provides labels for data that are generated from either benign or attack activities. To generate the annotation, we utilize the technical requirements defined in the cyberattack description as the ground truth. These requirements contain the necessary information (e.g., name of instances and artifacts) which are used as the key for automated data searching. Additionally, the chronological order of attack steps provides a natural arrangement for the annotated indicators. Therefore, PROVCON ensures that the extracted data are annotated with respect to the cyberattack description. This simplifies the attack investigation and alleviates the need for manual analysis.

IV. EXPERIMENTS AND RESULTS

This section demonstrates the effectiveness of PROVCON in constructing provenance graphs meant for augmenting SOC operations. We utilize the CTI reports from STIXnet [35] and AttacKG [33] as the basis for reproducing the cyberattack. The CTI reports are transformed into a structured and standardized representation called Structured Threat Information Expression (STIX) language [10]. By using CTI reports in STIX 2.1 format, we ensure standardization and compatibility with other systems while maintaining the shareability of cyberattack information.

A. Provenance Graph Construction

To reproduce a cyberattack, we first define the cyberattack description based on the primitives from the CTI report. To automate the transformation of domain-specific cyberattack knowledge into a structured description, we leverage the large language model's generative and understanding capabilities to devise the cyberattack description. Specifically, the framework uses in-context learning on ChatGPT-4 to learn about our DSL and the available primitives for building a cyber range. The cyberattack description for the respective attacks is then realized as a cyber range. Since the cyber range consists of heterogeneous instances, we categorize the data extraction process into two approaches. The first approach is to extract and transform the data from Linux-based instances, while the second approach is for Windows-based instances. For the first approach, we extract the sysdig logs from Linux-based instances and use them in our custom script to construct the Linux-based provenance graph. In the second approach, the Sysmon logs extracted from Windows-based instances are first loaded into Grafiki[3] for provenance graph visualization. We then use a custom script to extract the nodes and edges and construct the Windows-based provenance graph. The provenance graphs are saved as a graph description language in Graphviz DOT format. This format maintains compatibility with other systems and can be easily transformed into other graph-based formats.

For the experiments, we reproduce five APT campaigns and construct their respective provenance graphs as reflected in Table II. We reproduce the APT32 variant to demonstrate the additional insight that PROVCON can provide as described in § IV-B. Since every APT scenario is constructed with heterogeneous instances, the number of records from all instances is aggregated for the respective data type. Among the audit-based logs (i.e., auditd and sysdig for Linux, and Sysmon for Windows), a subset of the records are annotated as they are identified as key indicators. For the APT32 variant, there are no indicators found for both auditd and sysdig. This is because the *events* in the cyberattack description are entirely performed on Windows-based instances. As each provenance

[3]https://github.com/lucky-luk3/Grafiki

TABLE II: The statistics of the provenance data collected from reproduced cyberattacks.

APT	System Logs				Provenance Graph	Network Capture	Memory Dump
	Auditd	Sysdig	Sysmon Event	Others			
APT17	Records: 237598 Annotated: 3852	Records: 692854 Annotated: 2	Records: 1118 Annotated: 6	Records: 26010	Nodes: 1855 Edges: 4632	Records: 17459 File Size: 8.71 MB	File Size: 8363.61 MB
APT29	Records: 276178 Annotated: 3857	Records: 670692 Annotated: 2	Records: 1227 Annotated: 1	Records: 31243	Nodes: 2616 Edges: 7019	Records: 6266 File Size: 2.6 MB	File Size: 10431.48 MB
APT32	Records: 177916 Annotated: 1967	Records: 402263 Annotated: 4	Records: 4866 Annotated: 3	Records: 40753	Nodes: 2381 Edges: 6538	Records: 90413 File Size: 45.21 MB	File Size: 8363.61 MB
APT32 variant	Records: 190250 Annotated: 0	Records: 428313 Annotated: 0	Records: 3640 Annotated: 6	Records: 41859	Nodes: 4409 Edges: 8989	Records: 189624 File Size: 91.54 MB	File Size: 8363.61 MB
APT41	Records: 219507 Annotated: 535	Records: 695863 Annotated: 1	Records: 2651 Annotated: 33	Records: 35004	Nodes: 3618 Edges: 7888	Records: 8953 File Size: 2.39 MB	File Size: 12591.48 MB

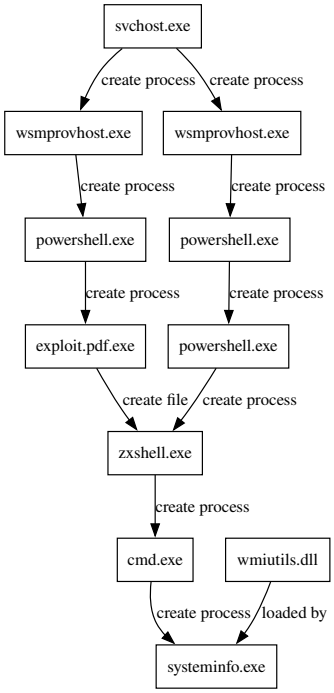

Fig. 2: Subset of provenance graph from APT17

graph is extracted from every instance, the aggregated number of nodes and edges from the provenance graphs of all instances are reflected in Table II. Aside from audit-based logs, we also extract other types of logs from the cyber range (e.g., application and authentication logs). Since PROVCON is designed to generate heterogeneous data as part of the dataset, we summarize the aggregated network packets and image size of the memory dump extracted from all of the instances in the cyber range.

For effective attack investigation, the provenance graph must include two types of information, namely causality and dependency. To illustrate the usefulness of provenance graphs constructed by PROVCON, we show a subset of APT17's provenance graph in Figure 2, which reflects the indicators found in the victim system. An example of causality is where `exploit.pdf.exe` causes the `zxshell.exe` file to be created. The causality information allows the analysts to perform root cause analysis using backtracking to identify an event that is responsible for the observed effect (e.g., `systeminfo.exe` process is caused by the execution of `exploit.pdf.exe`). Additionally, the execution of `zxshell.exe` process has a dependency on the file being created. The impact of creating the `zxshell.exe` file can be assessed using forward tracking to reveal the propagation of attack steps. Similar nodes can be grouped in the provenance graph by clustering them (e.g., `powershell.exe`). This way, analysts can identify and disrupt critical nodes to effectively halt the attack steps (e.g., disabling `powershell.exe` in Group Policy will prevent `exploit.pdf.exe` from being executed). This emphasizes the necessity of having a complete set of indicators in the provenance graph so that SOC can obtain high-quality insights to strategize their defenses effectively and allocate their resources appropriately.

B. Case Study: APT32 Campaign

This section demonstrates the use of PROVCON to enhance the understanding of existing cyberattacks by constructing provenance graphs. In this case study, we reproduce the APT32 campaign and extract insights that are useful for attack investigation. The APT32 threat actor, also known as OceanLotus, is a group known to be active since 2014. Their attack campaigns target public and private sectors, including industries from South East Asian countries. Their campaigns commonly involve spear-phishing, social engineering, and the use of full-featured malware in conjunction with commercial tools. The diversity and complexity of the APT32 campaign impose a challenge for analysts to acquire a comprehensive understanding of the cyberattack.

The first step to constructing the APT32's provenance graph is to utilize the information found in STIXnet's CTI report [35]. This is performed by identifying the relevant attack primitives from the STIX representation of the CTI report. An example of a primitive is a `tool` as reflected in Table III. This primitive describes the process of using COM scriptlets to download a payload. Based on the description provided, this primitive is classified under the *environment* category. This implies the need for an artifact that represents a downloader malware. Based on this description, we devise a script named `com_scriptlet.ps1` to represent this artifact. Due to the lack of information about the exact behavior of this downloader, we include a `ping` command as part of the script's execution. This is to ensure that network

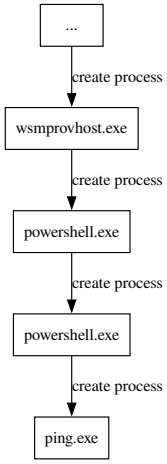

Fig. 3: Subset of provenance graph showing the execution of downloader

connectivity is established, as it is a prerequisite characteristic for a downloader malware.

The cyberattack description consists of technical requirements that are used to provision the resources and deploy the components to form the cyber range. In total, there are 13 artifacts, three instances, and two networks defined in the cyberattack description. The instances consist of a `C2Server`, a `VictimMachine`, and a networking node that provides connectivity across two networks. Every artifact is customized such that it exhibits the expected behavior when executed in the cyber range. While using the actual APT32 artifacts is ideal for replicating the cyberattack, this work focuses on ensuring that the key indicators are present in the constructed provenance graph. Therefore, we ensure that the information between the generated data and the CTI reports are consistent without any contradictions. Additionally, nine events are defined in the cyberattack description. Using the primitive above as an example, the cyberattack description defines the `com_scriptlet.ps1` as part of the `VictimMachine`. The `VictimMachine` then executes the PowerShell script during the attack. This way, the activities defined in the cyberattack description ensure that the key indicators are captured in the respective instances when executed.

The cyber range is deployed using the requirements defined in the cyberattack description. The *environment* components are deployed and configured to represent the infrastructure of the APT32 campaign. The activities defined in the *events* are chronologically executed throughout the infrastructure. The orchestration of activities replays the APT32 campaign in the cyber range, generating indicators that can be used for analysis. The data are extracted from the cyber range and transformed into a provenance graph. An example of the provenance graph obtained from the `VictimMachine` is reflected in Figure 3. The provenance graph indicates that a `powershell` initiates the `ping` program. However, the `com_scriptlet.ps1` script is not present in the provenance graph. Upon closer inspection of the Sysmon logs, the

TABLE III: A snippet of CTI report describing Cobalt Strike tool

Report	Property	Value
STIXnet [35]	Type	tool
	ID	tool–90...
	Name	Cobalt Strike
	Description	APT32 has used COM scriptlets to download Cobalt Strike beacons
	Tool Type	Exploitation
	Aliases	Beacon
AttacKG [33]	Type	tool
	ID	tool–23...
	Name	Cobalt Strike
	Description	Cobalt Strike stager used to download and execute shellcode from a remote server
	Tool Type	Exploitation

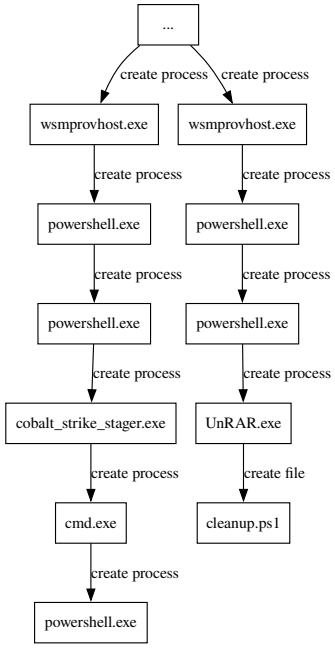

Fig. 4: Subset of provenance graph indicating Cobalt Strike tool execution

execution of `com_scriptlet.ps1` is obfuscated and is not captured during the construction of the provenance graph. Nevertheless, our annotation process captures the existence of `com_scriptlet.ps1` in the Sysmon logs. This is because all obfuscated strings are deobfuscated during the process of finding key indicators. The Sysmon logs also indicate that the `com_scriptlet.ps1`'s parent process ID belongs to the `powershell` process in the provenance graph. This further reinforces the identification of `ping` process as a key indicator that is initiated by the downloader malware.

Although the constructed APT32 provenance graph provides insight into the cyberattack, analysts are only limited to the

information made available from the generated indicators. This results in an incomplete understanding of APT32 as the indicators are derived from the perspective of one CTI report. In PROVCON, we provide the means for analysts to enhance their understanding by extending the cyberattack description with more primitives about the same APT32 campaign. In this case, we utilize a second APT32 CTI report from AttacKG [33] and reproduce the APT32 campaign on the cyber range. This CTI report includes the technicalities of the cyberattack from the victim's perspective, thus further completing the indicators about the APT32 campaign. To illustrate the enhancement, we show an example of a `tool` primitive in Table III. The primitive from both CTI reports describes the use of `Cobalt Strike` as a tool during the APT32 campaign. However, the second CTI report implies two malware (i.e., downloader and shellcode payload) and one instance (i.e., a remote server), thus providing a precise description that is relevant for constructing the *environment* in the cyber range. This shows that primitives from other CTI reports can be used to complement the initial cyberattack description. To demonstrate additional understanding about APT32, Figure 4 shows a subset of the provenance graph that illustrates the execution of the downloader malware. Two indicators that are defined in the cyberattack description are observed in the provenance graph, namely `cobalt_strike_stager.exe` and `cleanup.ps1`. The execution of `cobalt_strike_stager.exe` creates additional processes that are used to execute future commands (i.e., `cmd.exe` and `powershell.exe`). Additionally, the provenance graph also indicates the creation of the `cleanup.ps1` script from `UnRAR.exe`. This PowerShell script is used as a tool to remove attack traces from Sysmon logs at the end of the attack events. Based on the additional information found in the provenance graph, analysts can obtain new insights about the cyberattack, thus advancing toward a comprehensive and up-to-date understanding of the APT32 campaign.

V. DISCUSSION

The use of CTI reports as the basis in PROVCON allows analysts to construct a provenance graph with indicators from documented observations. This allows PROVCON to complement existing knowledge by reproducing cyberattacks to generate relevant data. As a result, the generated insights that originate from documented perspectives are useful to augment the current knowledge of cyberattacks. Furthermore, PROVCON design enables the analyst to halt the execution of attack activities midway, allowing them to perform cyberattack introspection. Additionally, the segregation of primitives between *environment* and *events* allows PROVCON to execute the attack activities from a specific cyberattack (e.g., APT32 campaign) on a different set of computing environments. Therefore, SOC can analyze and observe the impact of a particular cyberattack when executed on their environment. This creates a provenance graph according to their cyber infrastructure, yielding personalized insights that are not available in existing attack datasets. Hence, SOC can align their ongoing workflow of cyberattack monitoring and detection systems with the insights from PROVCON, thus leading to the overall improvement of their cybersecurity posture.

Despite the need for seamless construction of provenance graphs, an existing challenge hinders end-to-end automation for PROVCON. The actual artifacts (e.g., malware, exploits) that are involved in the cyberattack may not be readily available. Even when they are publicly available, they may not be executable in our cyber range (e.g., missing dependencies, obfuscation techniques). To overcome this challenge, one method is to design an artifact generation tool to automatically reproduce disarmed artifacts that exhibit the same behavior and indicators according to a standardized knowledge base (e.g., MITRE Malware Behavior Catalog (MBC) [42], ATT&CK framework [49]). In this work, we devise artifacts aimed at displaying indicators that should be present in the resulting provenance graph. We ensure that the custom artifacts are non-malicious and only executed within the boundary of our cyber range. Therefore, PROVCON is described as semi-automated and is progressing towards the goal of fully automated insight generation.

VI. RELATED WORK

Cyberattack data generation is a crucial aspect of cyberattack detection and investigation, as it provides a comprehensive view of the attack activities. Existing data generation works mainly focus on generating provenance graphs from attack simulations [5], [6], [30], [47] or generating synthetic provenance graphs from models [14], [20], [21], [36].

Attack Simulations – Flurry [30] is a framework that simulates attacks and benign activities on hosts and generates provenance graphs for representation learning. For Operational Technology (OT) systems, a work [47] presents a modular dataset generation framework for Supervisory Control and Data Acquisition (SCADA) cyberattacks to aid the development of attack datasets.

Data Synthesis – ProvGen [21] is a generator aimed at producing large synthetic provenance graphs with predictable properties of arbitrary size. To address the unbalanced attack dataset used in model training, a work [36] proposes a framework for generating cyberattack data using Generative Adversarial Network (GAN) to expand existing cyberattack datasets.

Although these works can generate provenance graphs, they only support limited attack and benign activities and cannot reflect the complexity of real-world cyberattacks. PROVCON leverages the CTI report to gather a comprehensive set of cyberattack primitives and construct provenance graphs in complex cyber environments that closely resemble real-world cyberattacks.

VII. CONCLUSION

We highlight the need for SOC to utilize provenance graphs that contain comprehensive and timely information about existing cyberattacks. This work proposes a framework to construct a provenance graph that provides useful insights for SOC

operations. The framework utilizes the cyberattack primitives extracted from CTI reports to reproduce the cyberattack in a cyber range. The data extracted from the cyber range are used to construct provenance graphs which contains key indicators that are representative of the cyberattack. The insights obtained from the provenance graphs allow analysts to deepen their understanding of the cyberattack, thus complementing the ongoing operations in SOC. To support attack analysis in SOC, we aim to keep our repository updated with data generated from future cyberattack reproductions.

ACKNOWLEDGMENT

This research is supported by the National Research Foundation, Singapore, through the National Cybersecurity R&D Lab at the National University of Singapore under its National Cybersecurity R&D Programme (Award No. NCR25-NCL P3-0001). 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] "DARPA Transparent Computing Engagement 3," 2018, https://github.com/darpa-i2o/Transparent-Computing/blob/master/README-E3.md.

[2] "DARPA Operationally Transparent Cyber (OpTC) Data," 2020, https://github.com/FiveDirections/OpTC-data.

[3] "DARPA Transparent Computing Engagement 5," 2020, https://github.com/darpa-i2o/Transparent-Computing.

[4] "Dtrace on freebsd," 2022, https://wiki.freebsd.org/DTrace/.

[5] Y. Al-Hadhrami and F. K. Hussain, "Real time dataset generation framework for intrusion detection systems in iot," *Future Generation Computer Systems*, vol. 108, pp. 414–423, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X19322678

[6] F. A. Alhaidari and A. M. Alrehan, "A simulation work for generating a novel dataset to detect distributed denial of service attacks on vehicular ad hoc network systems," *International Journal of Distributed Sensor Networks*, vol. 17, no. 3, p. 15501477211000287, 2021. [Online]. Available: https://doi.org/10.1177/15501477211000287

[7] A. Alsaheel, Y. Nan, S. Ma, L. Yu, G. Walkup, Z. B. Celik, X. Zhang, and D. Xu, "ATLAS: A sequence-based learning approach for attack investigation," in *Proceedings of the USENIX Security Symposium (USENIX Security)*, 2021, pp. 3005–3022.

[8] E. Altinisik, F. Deniz, and H. T. Sencar, "Provg-searcher: A graph representation learning approach for efficient provenance graph search," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 2247–2261. [Online]. Available: https://doi.org/10.1145/3576915.3623187

[9] O. Bajaber, B. Ji, and P. Gao, "P4control: Line-rate cross-host attack prevention via in-network information flow control enabled by programmable switches and ebpf," in *2024 IEEE Symposium on Security and Privacy (SP)*, 2024, pp. 4610–4628.

[10] S. Barnum, "Standardizing Cyber Threat Intelligence Information with the Structured Threat Information eXpression (STIX™)," Tech. Rep., 2014. [Online]. Available: http://stixproject.github.io/getting-started/whitepaper/

[11] B. Bhattarai and H. Huang, "Steinerlog: Prize collecting the audit logs for threat hunting on enterprise network," in *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, ser. ASIA CCS '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 97–108.

[12] T. Chen, Q. Song, X. Qiu, T. Zhu, Z. Zhu, and M. Lv, "Kellect: a kernel-based efficient and lossless event log collector for windows security," 2023. [Online]. Available: https://arxiv.org/abs/2207.11530

[13] Z. Cheng, Q. Lv, J. Liang, Y. Wang, D. Sun, T. Pasquier, and X. Han, "Kairos: Practical intrusion detection and investigation using whole-system provenance," in *2024 IEEE Symposium on Security and Privacy (SP)*, 2024, pp. 3533–3551.

[14] C. G. Cordero, E. Vasilomanolakis, A. Wainakh, M. Mühlhäuser, and S. Nadjm-Tehrani, "On generating network traffic datasets with synthetic attacks for intrusion detection," *ACM Trans. Priv. Secur.*, vol. 24, no. 2, Jan. 2021. [Online]. Available: https://doi.org/10.1145/3424155

[15] H. Ding, J. Zhai, D. Deng, and S. Ma, "The case for learned provenance graph storage systems," in *32nd USENIX Security Symposium (USENIX Security 23)*. Anaheim, CA: USENIX Association, Aug. 2023, pp. 3277–3294. [Online]. Available: https://www.usenix.org/conference/usenixsecurity23/presentation/ding-hailun-provenance

[16] H. Ding, J. Zhai, Y. Nan, and S. Ma, "Airtag: towards automated attack investigation by unsupervised learning with log texts," in *Proceedings of the 32nd USENIX Conference on Security Symposium*, ser. SEC '23. USA: USENIX Association, 2023.

[17] DOMARS, "Event tracing for windows (etw)," 2021, https://learn.microsoft.com/en-us/windows-hardware/drivers/devtest/event-tracing-for-windows–etw-.

[18] F. Dong, S. Li, P. Jiang, D. Li, H. Wang, L. Huang, X. Xiao, J. Chen, X. Luo, Y. Guo, and X. Chen, "Are we there yet? an industrial viewpoint on provenance-based endpoint detection and response tools," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2023.

[19] P. Fang, P. Gao, C. Liu, E. Ayday, K. Jee, T. Wang, Y. F. Ye, Z. Liu, and X. Xiao, "Back-Propagating system dependency impact for attack investigation," in *Proceedings of the USENIX Security Symposium (USENIX Security)*, 2022, pp. 2461–2478.

[20] A. Ferriyan, A. H. Thamrin, K. Takeda, and J. Murai, "Generating network intrusion detection dataset based on real and encrypted synthetic attack traffic," *Applied Sciences*, vol. 11, no. 17, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/17/7868

[21] H. Firth and P. Missier, "Provgen: Generating synthetic prov graphs with predictable structure," in *Provenance and Annotation of Data and Processes*, B. Ludäscher and B. Plale, Eds. Cham: Springer International Publishing, 2015, pp. 16–27.

[22] Fortra, "Cobalt strike," 2024, https://www.cobaltstrike.com/.

[23] A. Goyal, X. Han, G. Wang, and A. Bates, "Sometimes, you aren't what you do: Mimicry attacks against provenance graph host intrusion detection systems," in *30th Network and Distributed System Security Symposium*, 2023.

[24] A. Goyal, G. Wang, and A. Bates, "R-caid: Embedding root cause analysis within provenance-based intrusion detection," in *2024 IEEE Symposium on Security and Privacy (SP)*, 2024, pp. 3515–3532.

[25] X. Han, T. Pasquier, A. Bates, J. Mickens, and M. Seltzer, "Unicorn: Runtime provenance-based detector for advanced persistent threats," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2022.

[26] M. N. Hossain, S. Sheikhi, and R. Sekar, "Combating dependence explosion in forensic analysis using alternative tag propagation semantics," in *Proceedings of the IEEE Symposium on Security and Privacy (IEEE S&P)*, 2020, pp. 1139–1155.

[27] M. Inam, Y. Chen, A. Goyal, J. Liu, J. Mink, N. Michael, S. Gaur, A. Bates, and W. U. Hassan, "SoK: History is a vast early warning system: Auditing the provenance of system intrusions," in *Proceedings of the IEEE Symposium on Security and Privacy (IEEE S&P)*, 2023, pp. 307–325.

[28] B. Jacob, P. Larson, B. Leitao, and S. Da Silva, "Systemtap: instrumenting the linux kernel for analyzing performance and functional problems," *IBM Redbook*, vol. 116, 2008.

[29] Z. Jia, Y. Xiong, Y. Nan, Y. Zhang, J. Zhao, and M. Wen, "MAGIC: Detecting advanced persistent threats via masked graph representation learning," in *33rd USENIX Security Symposium (USENIX Security 24)*. Philadelphia, PA: USENIX Association, Aug. 2024, pp. 5197–5214. [Online]. Available: https://www.usenix.org/conference/usenixsecurity24/presentation/jia-zian

[30] M. Kapoor, J. Melton, M. Ridenhour, T. Moyer, and S. Krishnan, "Flurry: A fast framework for provenance graph generation for representation learning," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, ser. CIKM '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 4887–4891. [Online]. Available: https://doi.org/10.1145/3511808.3557200

[31] J. Li, R. Zhang, J. Liu, and G. Liu, "Logkernel: A threat hunting approach based on behaviour provenance graph and graph kernel clustering," *Security and Communication Networks*, vol. 2022, no. 1, p. 4577141, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1155/2022/4577141

[32] S. Li, F. Dong, X. Xiao, H. Wang, F. Shao, J. Chen, Y. Guo, X. Chen, and D. Li, "Nodlink: An online system for fine-grained apt attack detection and investigation," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2024.

[33] Z. Li, J. Zeng, Y. Chen, and Z. Liang, "Attackg: Constructing technique knowledge graph from cyber threat intelligence reports," in *Computer Security – ESORICS 2022: 27th European Symposium on Research in Computer Security, Copenhagen, Denmark, September 26–30, 2022, Proceedings, Part I*. Berlin, Heidelberg: Springer-Verlag, 2022, p. 589–609.

[34] E. Manzoor, S. M. Milajerdi, and L. Akoglu, "Fast memory-efficient anomaly detection in streaming heterogeneous graphs," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1035–1044. [Online]. Available: https://doi.org/10.1145/2939672.2939783

[35] F. Marchiori, M. Conti, and N. V. Verde, "Stixnet: A novel and modular solution for extracting all stix objects in cti reports," in *Proceedings of the 18th International Conference on Availability, Reliability and Security*, ser. ARES '23. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: https://doi.org/10.1145/3600160.3600182

[36] T. Merino, M. Stillwell, M. Steele, M. Coplan, J. Patton, A. Stoyanov, and L. Deng, *Expansion of Cyber Attack Data from Unbalanced Datasets Using Generative Adversarial Networks*. Cham: Springer International Publishing, 2020, pp. 131–145. [Online]. Available: https://doi.org/10.1007/978-3-030-24344-9_8

[37] Metasploit, "Metasploit," 2024, https://github.com/rapid7/metasploit-framework.

[38] Microsoft, "Security auditing," 2021, https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-10/security/threat-protection/auditing/security-auditing-overview.

[39] S. M. Milajerdi, B. Eshete, R. Gjomemo, and V. Venkatakrishnan, "POIROT: Aligning attack behavior with kernel audit records for cyber threat hunting," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2019, p. 1795–1812.

[40] ——, "TREC: APT tactic / technique recognition via few-shot provenance subgraph learning," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2024.

[41] S. M. Milajerdi, R. Gjomemo, B. Eshete, R. Sekar, and V. Venkatakrishnan, "HOLMES: real-time APT detection through correlation of suspicious information flows," in *Proceedings of the IEEE Symposium on Security and Privacy (IEEE S&P)*, 2019, pp. 1137–1152.

[42] MITRE, "Malware Behavior Catalog." [Online]. Available: https://github.com/MBCProject/mbc-markdown

[43] Mozilla, "Firefox os/deviceqa/collectlogs," 2015, https://wiki.mozilla.org/Firefox_OS/DeviceQA/CollectLogs.

[44] K. Mukherjee, J. Wiedemeier, T. Wang, J. Wei, F. Chen, M. Kim, M. Kantarcioglu, and K. Jee, "Evading {Provenance-Based}{ML} detectors with adversarial system actions," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 1199–1216.

[45] T. Pasquier, X. Han, M. Goldstein, T. Moyer, D. Eyers, M. Seltzer, and J. Bacon, "Practical whole-system provenance capture," in *Proceedings of the 2017 Symposium on Cloud Computing*, 2017, pp. 405–418.

[46] Redhat, "The linux audit framework," 2017, https://github.com/linux-audit/.

[47] N. R. Rodofile, K. Radke, and E. Foo, "Framework for scada cyber-attack dataset creation," in *Proceedings of the Australasian Computer Science Week Multiconference*, ser. ACSW '17. New York, NY, USA: Association for Computing Machinery, 2017. [Online]. Available: https://doi.org/10.1145/3014812.3014883

[48] M. Sharif, P. Datta, A. Riddle, K. Westfall, A. Bates, V. Ganti, M. Lentzk, and D. Ott, "Drsec: Flexible distributed representations for efficient endpoint security," in *2024 IEEE Symposium on Security and Privacy (SP)*, 2024, pp. 3609–3624.

[49] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "Mitre att&ck: Design and philosophy," in *Technical report*. The MITRE Corporation, 2018.

[50] Sysdig, "Sysdig," 2017, https://sysdig.com/.

[51] A. R. Team, "Atomic Red Team," 2021, https://github.com/redcanaryco/atomic-red-team.

[52] M. Ur Rehman, H. Ahmadi, and W. Ul Hassan, "Flash: A comprehensive approach to intrusion detection via provenance graph representation learning," in *2024 IEEE Symposium on Security and Privacy (SP)*, 2024, pp. 3552–3570.

[53] Z. Xu, P. Fang, C. Liu, X. Xiao, Y. Wen, and D. Meng, "Graph summarization on system audit logs for attack investigation," in *Proceedings of the IEEE Symposium on Security and Privacy (IEEE S&P)*, 2022.

[54] F. Yang, J. Xu, C. Xiong, Z. Li, and K. Zhang, "PROGRAPHER: An anomaly detection system based on provenance graph embedding," in *32nd USENIX Security Symposium (USENIX Security 23)*. Anaheim, CA: USENIX Association, Aug. 2023, pp. 4355–4372. [Online]. Available: https://www.usenix.org/conference/usenixsecurity23/presentation/yang-fan

[55] J. Zeng, Z. L. Chua, Y. Chen, K. Ji, Z. Liang, and J. Mao, "Watson: Abstracting behaviors from audit logs via aggregation of contextual semantics," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2021.

[56] J. Zengy, X. Wang, J. Liu, Y. Chen, Z. Liang, T.-S. Chua, and Z. L. Chua, "ShadeWatcher: Recommendation-guided cyber threat analysis using system audit records," in *Proceedings of the IEEE Symposium on Security and Privacy (IEEE S&P)*, 2022, pp. 489–506.

[57] M. Zipperle, F. Gottwalt, E. Chang, and T. Dillon, "Provenance-based intrusion detection systems: A survey," *ACM Comput. Surv.*, vol. 55, no. 7, Dec. 2022.