Resilience Against APTs: A Provenance-based IIoT Dataset for Cybersecurity Research

Erfan Ghiasvand[1], Suprio Ray[1], Shahrear Iqbal[2], Sajjad Dadkhah[1], and Ali A. Ghorbani[1]

[1] Faculty of Computer Science, University of New Brunswick (UNB), Fredericton, NB, Canada {eghiasva,sray,sdadkhah,ghorbani}@unb.ca
[2] National Research Council, Fredericton NB, Canada
shahrear.iqbal@nrc-cnrc.gc.ca

Abstract. The Industrial Internet of Things (IIoT) is a transformative paradigm that integrates smart sensors, advanced analytics, and robust connectivity within industrial processes, enabling real-time data-driven decision-making and enhancing operational efficiency across diverse sectors, including manufacturing, energy, and logistics. IIoT is susceptible to various attack vectors, with Advanced Persistent Threats (APTs) posing a particularly grave concern due to their stealthy, prolonged, and targeted nature. The effectiveness of machine learning-based intrusion detection systems in APT detection has been documented in the literature. However, existing cybersecurity datasets often lack crucial attributes for APT detection in IIoT environments.

Incorporating insights from prior research on APT detection using provenance data and intrusion detection within IoT systems, we present the CICAPT-IIoT dataset. The main goal of this paper is to propose a novel APT dataset in the IIoT setting that includes essential information for the APT detection task. In order to achieve this, a testbed for IIoT is developed, and over 20 attack techniques frequently used in APT campaigns are included. The performed attacks create some of the invariant phases of the APT cycle, including Data Collection and Exfiltration, Discovery and Lateral Movement, Defense Evasion, and Persistence. By integrating network logs and provenance logs with detailed attack information, the CICAPT-IIoT dataset presents foundation for developing holistic cybersecurity measures. Additionally, a comprehensive dataset analysis is provided, presenting cybersecurity experts with a strong basis on which to build innovative and efficient security solutions.

Keywords: Industrial IoT · Advanced Persistent Threats · Data Provenance · Self-Supervised Learning.

1 Introduction

Advanced Persistent Threats (APTs) represent a sophisticated category of cyberattacks, where an unauthorized user gains access to a network and remains undetected for a long period of time. Some attackers aim to harm organizations for financial motives or to gain notoriety by damaging a company's reputation, and they do not conceal their actions. However, in recent years, another type of attacker group has risen in prominence, which is characterized by a deliberate and methodical approach. They employ a "low and slow" strategy with the goal of either stealing sensitive data from their targets or disrupting their operations [4]. APTs represent a significant threat to critical infrastructure systems and have been responsible for numerous severe incidents. APT attacks are distinguished from typical cyberattacks by some key characteristics, such as complexity, persistence, being targeted, and elusiveness. APT attacks typically consist of several distinct phases, each with specific objectives and strategies. While the exact phases can vary depending on the attack group and campaign, the following [32] are common phases in an APT attack: (1) Initial Compromise, (2) Establishing a Foothold, (3) Privilege Escalation, (4) Reconnaissance, (5) Lateral Movement, (6) Maintaining Persistence, and (7) Data Collection and Exfiltration.

Indeed, Industrial Internet of Things (IIoT) networks represent a particularly vulnerable target for APT attacks. Originally centered around general applications, IoT has extended its influence to diverse sectors, including industry, where there is an increasing drive to interconnect previously isolated components, facilitating both intra-component communication and connections to the broader Internet [30]. Industrial IoT enables the seamless integration of several devices with sensing, identification, processing, communication, and networking capabilities [14]. Researchers use many system architectures for IIoT systems, such as Brown-IIoTbed [1], to develop an IIoT environment.

The security and safety of IIoT systems have been the subject of substantial research due to the essential importance and sensitivity of Industrial IoT applications. As demonstrated by historical occurrences like Stuxnet [29], the Ukrainian power plant attacks [45], and the TRITON incident [15], attacks on the IIoT can have significant consequences that go beyond the scope of a company's operations and may compromise the safety of citizens and even the entire nation. Research findings on the security of Industrial IoT reveal the disturbing fact that IIoT devices are susceptible to weaknesses, as described in [46] and [43]. This paints an alarming picture of the security environment used in current IIoT applications.

The convergence of critical infrastructure, interconnected devices, and often limited security measures within IIoT environments makes them an attractive and high-impact target for sophisticated and persistent adversaries like APT groups. These attackers seek to exploit vulnerabilities within IIoT systems to achieve their objectives, which can have significant consequences for industrial operations and, in some cases, national security. As a result, safeguarding IIoT networks from APT attacks is crucial in the realm of cybersecurity.

Traditional threat detection systems, including signature-based and anomaly-based approaches, face limitations in effectively detecting long-running APT campaigns [12]. Signature-based systems struggle to detect APTs that leverage zero-day exploits and new vulnerabilities [18]. Conversely, anomaly-based systems, that leverage network logs [48], system calls [11], and related system events

[47], often encounter difficulties in modeling extended system behavior patterns. These systems are also vulnerable to evasion techniques since they primarily examine short sequences of system calls and events, thus limiting their ability to uncover sophisticated APT activities.

According to recent studies [10,23,8,25], data provenance may be a more reliable data source for identifying APTs. Data provenance depicts the flow of information between system entities, such as processes, and objects, such as files and sockets, as a directed acyclic graph (DAG), shows how a system is being used. Even when events are separated in time, this representation links the graph's causally connected events. Consequently, despite APT-affected systems often mimicking normal system behavior, the wealth of contextual information inherent in provenance data enhances the ability to distinguish between benign and malicious events [49]. Despite the demonstrated efficacy of utilizing provenance data in detecting APT attacks, researchers in the field encounter a significant challenge: the scarcity of available datasets. Moreover, the existing datasets frequently do not cover APT scenarios in IIoT environments, making it even more challenging to explore this problem. In addition to that, the APT detection methods currently proposed often lack compatibility with some features of the ever-changing APT landscape.

In this research, we introduce CICAPT-IIoT[3], an APT attack dataset developed within an IIoT environment, to assist researchers in security analysis and developing detection methods. To achieve this, an IIoT testbed was established in a semi-controlled setting, mirroring real-world industrial operations. A realistic APT scenario, containing key APT phases like data exfiltration and defense evasion, was then implemented and executed. Raw and processed data collected during this scenario are also made available, enabling researchers to utilize and derive new features for enhanced security insights on using provenance data for the APT detection task. We also develop a self-supervised learning (SSL) model to process provenance data for APT detection tasks. This model is specifically designed to be compatible with the unique features of APT attacks and the heterogeneous nature of the provenance graphs.

The main contributions of our research are as follows:

- We introduce the CICAPT-IIoT dataset, a novel and comprehensive APT attack dataset captured within the IIoT environment. This dataset is generated using a hybrid testbed consisting of real and simulated IIoT components to demonstrate the complexity and diversity of modern technology systems;
- The dataset contains more than 20 distinct attack techniques divided into eight main attack tactics that map into the APT attack scenarios, inspired by the APT29 [33] campaigns. This APT scenario enhances the dataset's effectiveness in APT detection research;
- To evaluate the effectiveness of machine learning algorithms in APT-detection tasks, we applied several ML models on the CICAPT-IIoT dataset and analyzed their performance. Our evaluation uses a provenance-based detection framework offering insights into the practical challenges and considerations

[3] Canadian Institute for Cybersecurity Advanced Persistent Threats Dataset for IIoT

in deploying machine learning solutions for APT detection in the IIoT landscape;
- We propose a self-supervised based method and use the CICAPT-IIoT dataset to test and evaluate its performance in provenance-based APT detection. Our results show the effectiveness of the SSL-based model for the provenance graphs.

The rest of this paper is organized as follows. We discussed the related works in Section 2. Section 3 provides an overview of the testbed, its different components, and the APT attack emulation plan. we also explain the dataset generation experiments and the dataset properties in this section. Furthermore, a thorough analysis of the dataset is presented in the section 4. Next, we describe predictive models for APT detection task in section 5 and evaluate these models performance using CICAPT-IIoT dataset in section 6. Finally, Section 7 presents the conclusion of this research.

2 Related Works

Recent research has explored the utility of provenance data across various domains, including security, reproducibility, data trustworthiness, and intrusion detection [28]. These studies have demonstrated its potential for improving the dependability and security of information systems against various threats such as APTs [31]. Datasets play a crucial role in any attack detection research, enabling the study and modeling of behavior to identify attack activities [44]. However, the majority of available datasets are generated for conventional intrusion detection rather than detecting Advanced Persistent Threats. Such datasets often lack the complexity inherent in APT cycle phases and typically comprise only network or system logs. In this section, we provide an overview of the datasets currently used in the literature for APT detection and general attack detection. Additionally, we offer a brief review of the literature on provenance data, methods for capturing provenance, and provenance-based attack detection techniques.

2.1 Related Datasets

The significance of datasets in attack detection research cannot be overstated, as they are fundamental to the development, testing, and refinement of detection algorithms. High-quality datasets provide a realistic representation of valuable data such as system logs and network traffic, and include both benign activities and malicious attacks, that are crucial for training and evaluating intrusion detection systems.

The TON IoT dataset[3] includes telemetry data from IoT/IIoT services and contains network traffic collected from a realistic representation of a medium-scale network at an IoT Lab. This dataset includes a variety of cyberattacks, including scanning attacks, Denial of Service (DoS) attacks, ransomware, and Man-In-The-Middle (MITM) attacks, among others, providing researchers with a comprehensive resource to study, understand, and develop countermeasures against these threats. The dataset is designed for multi-classification problems, incorporating labels for normal and attack classes, and sub-classes of attacks targeting IoT/IIoT applications. DAPT 2020 [34], is a benchmark dataset specifically designed to address the challenges in modeling and detecting APTs. This dataset includes attacks that are hard to distinguish from normal traffic flows and encompass both public-to-private interface traffic and internal network traffic. The APT stages that this dataset covers are Reconnaissance, Foothold Establishment, Lateral Movement, and Data Exfiltration which are all crucial steps in APT campaigns.

X-IIoTID [2] is a dataset for intrusion detection in the Industrial Internet of Things (IIoT) environment. IIoT systems, due to their vast connectivity and deployment of various protocols and devices, present significant security challenges. This dataset is designed to be both connectivity-agnostic and device-agnostic, thereby suitable for the heterogeneous and interoperable nature of IIoT environments. The authors state that X-IIoTD covers the Reconnaissance, Weaponization, C&C, and Lateral movement stages of an attack scenario. Edge-IIoTset [16], is a cybersecurity dataset designed for IoT and IIoT applications, useful for both centralized and federated learning intrusion detection systems. The dataset is generated from a custom-built IoT/IIoT testbed incorporating a wide range of devices, sensors, protocols, and cloud/edge configurations. Edge-IIoTset includes over 10 types of IoT devices that generate various types of data, such as temperature, humidity, and ultrasonic sensor readings, and contains data related to DoS, DDoS, MitM, Reconnaissance, and malware attacks.

The DARPA OpTC dataset [17] contains data from a pilot study aimed at testing the scalability of DARPA Transparent Computing technologies for cyber defense. This dataset, generated during a two-week evaluation in a highly instrumented environment, capturing both benign activities and malware injections across one thousand Windows 10 endpoints and serves as a critical resource for analyzing the effectiveness of scaled cyber defense technologies in detecting APTs within large-scale network environments. CICIoT2023 [36] is an IoT attack dataset designed to aid in the development of security analytics applications for real IoT operations by executing 33 attacks within an IoT topology of 105 devices, classifying these attacks into seven categories: DDoS, DoS, Recon, Web-based, brute force, spoofing, and Mirai, all executed by malicious IoT devices targeting other IoT devices.

Unraveled [35], one of the most recent datasets is a semi-synthetic dataset crafted to emulate APT attacks. In response to the scarcity of publicly accessible APT datasets, the creators endeavored to enrich this dataset with a range of sophisticated attack scenarios derived from the MITRE ATT&CK database. Additionally, they designed an Employee Behavior Generation model aimed at replicating typical employee activities. The dataset is collected during a 6-week period and contains data from Reconnaissance, Foothold Establishment, Lateral movement, and Data exfiltration stages of APTs.

2.2 Provenance Data and Provenance-based Attack Detection

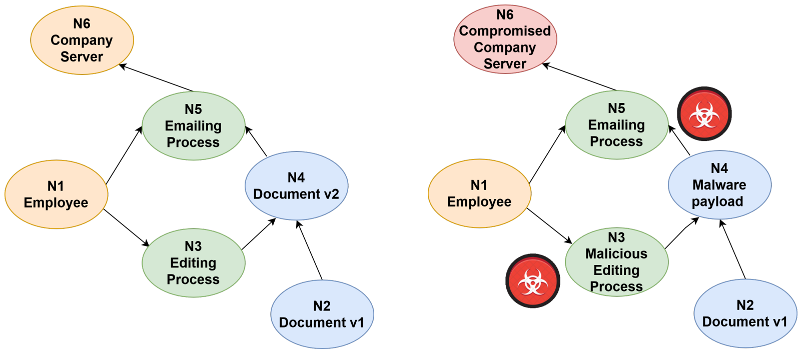

Fig. 1. Sample provenance graphs showing a benign scenario (Left) and an attack scenario (Right)

Data provenance refers to the documentation or record of the origin, lineage, and history of data. It includes every step of the creation, modification, and evolution of data over time [24]. Tracking the origins of data, such as where and how it was created and its evolution through various processing stages and transfers, are components of data provenance.
Bates et al. [9] introduced the Linux Provenance Modules (LPM), a framework designed for secure provenance collection on Linux operating systems. CamFlow [40], using a similar architecture, implemented a practical whole-system provenance system. This system leverages the Linux Security Module (LSM) and NetFilter hooks, capturing provenance data within the Linux environment. Some researchers developed cross-platform data provenance platforms that can collect provenance on different operating systems. For example SPADE [19] is a provenance solution capable of tracking and analyzing provenance from multiple possibly distributed sources including OS's auditing mechanisms.There has also been some research to introduce the concept of provenance to the IoT world and use its benefits to mitigate IoT security challenges [26]. Researchers in [6,5,38,42] have proposed various methods to collect and use provenance data in the IoT environment.
Since provenance data provides a thorough history and origin of any information within a system, it has become a valuable tool in intrusion detection systems. Cybersecurity researchers have explored provenance potential to improve security systems [39]. Works such as UNICORN [22] and ANUBIS [7] have utilized provenance data to train anomaly detection models for identifying APT activities within target environments.

UNICORN is an anomaly-based detection system designed to mitigate APTs that utilize data provenance by transforming system execution information into a directed acyclic graph (DAG). UNICORN employs graph sketching to create a scalable, incrementally updatable, fixed-size data structure. ANUBIS, another provenance-based framework for APT detection, is a machine learning-based system that uses a Bayesian Neural Network (BNN) for the classification of the system's events. This allows ANUBIS not only to detect APTs with high accuracy but also to explain its predictions, making it a valuable tool for cyber-response teams.

Figure 1 presents two different provenance graphs. The graph on the left depicts a benign scenario where a document is accessed, edited by a user, and subsequently sent to the company server. The graph on the right illustrates a simplified APT attack scenario: within a compromised system, a malicious editing process attaches a malicious payload to the file, which is then transmitted to the server, resulting in its compromise.

3 CICAPT-IIoT - A Semi-Synthetic IIoT APT Dataset

In this section, we present an overview of our data collection setup and the main components of our system. As APT detection research often suffers from the lack of realistic, open-source datasets, our research involves developing CICAPT-IIoT, a semi-synthetic dataset that imitates the characteristics of APT behaviors. The dataset generation design comprises diverse tools and devices, making it possible to gather a suitable dataset for APT detection within IIoT systems. Here we describe the data collection procedure and the various phases of the data generation process. Next, we explore our attack emulation plan and its different steps. Finally, we analyze the dataset and discuss the techniques we've used to distinguish between malicious and benign data.

3.1 System Overview

We have developed a simulation testbed to have a controlled environment that is useful for IIoT research and particularly beneficial for simulating APT scenarios. This testbed is based on the architecture of the Brown-IIoTbed framework [1]. Our testbed's structure integrates various virtual and physical components to mirror the complexity and interactions of real-world IIoT systems. Figure 2 illustrates an overview of the system.

At the heart of our testbed is the NS3 network simulator [37], running on an Ubuntu host. The NS3 is essential in bridging actual and simulated nodes through its tap bridge module, allowing us to coordinate a seamless integration between real and virtual network components. The testbed setup involves two Ubuntu VMs and two Kali Linux VMs hosted on a machine running NS3. Ubuntu VM1 acts as the gateway, managing network traffic, and is equipped with tools like Auditd and SPADE for logging and translating logs into provenance data, respectively. It also subscribes to MQTT topics, playing a critical role within the testbed's MQTT ecosystem. Ubuntu VM2 functions as an MQTT publisher and SCADA system via SCADABR software, interacting with a PLC simulated

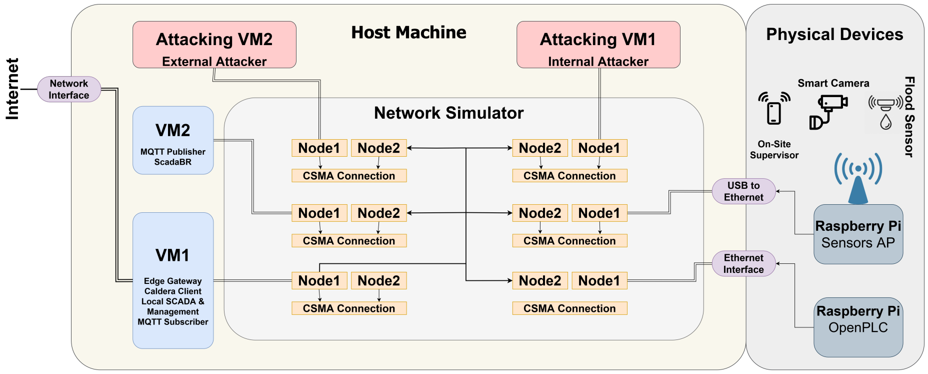

Fig. 2. Overview of the testbed

on Raspberry Pi1, which operates with OpenPLC and communicates using the Modbus protocol. Raspberry Pi2 serves as a WiFi access point, enhancing the network's connectivity to IoT sensors. Kali VM1 and VM2 are set up as internal and external threat actors, equipped with MITRE Caldera and various attack tools, respectively. This sophisticated simulation environment not only generates comprehensive data including system logs, network traffic, and provenance data essential for APT research in IIoT environments but also mimics real IIoT operations to facilitate advanced IIoT security research. Adding devices like cameras, leakage sensors, and PLCs and using protocols such as MODBUS and MQTT helps the testbed more accurately replicate an IIoT environment [27]. PLCs enable realistic automation and control simulations typical in industrial settings, and MODBUS and MQTT are common device-to-device communication protocols in IIoT systems. Table 1 shows the testbed components and their roles in the experiments.

Table 1. List of the testbed components

Device	Role
Ubuntu VM1	Gateway Local Management MQTT Subscriber
Ubuntu VM2	MQTT Publisher ScadaBR
Kali VM1	Internal Attacker Caldera Server
Kali VM2	External Attacker
Raspberry Pi1	OpenPLC
Raspberry Pi2	WiFi Access Point
Litokam Smart Camera	Camera
ConnectifyFlood Sensor	Flood Sensor

3.2 Attack Emulation Plan

APT29, also known as ‘The Dukes’ or ‘Cozy Bear’, is a sophisticated cyber threat group noted for its advanced cyber espionage tactics and persistent attacks. The group’s activities have had significant impacts, leading MITRE to publish an adversary emulation plan for APT29 within the framework of MITRE Caldera [13]. This emulation plan, however, uses tactics and techniques mostly tailored for Windows environments, which are not directly transferable to Linux systems. Therefore, we adapted APT29 emulation plan and the MITRE ATT&CK framework’s Tactics, Techniques, and Procedures (TTPs) to design a customized attack emulation plan suitable for the Linux-based testbed. This plan aims to replicate APT29’s operational patterns within the unique environment of the testbed. Table 2 shows the different APT tactics and techniques used in the dataset. The developed emulation plan encompasses several stages, each reflecting a typical phase in an APT campaign, including Data Collection and Exfiltration, Deployment of Stealth Toolkits for further activities, Defense and Discovery Evasion, Maintaining Persistence, Accessing System Credentials, and Lateral Movement to other components in the network.

3.3 Data Collection and Experiments

The NS3 simulator serves as the primary platform for our testbed, managing network connections and enabling the monitoring and logging of all network packets. It operates in Real-Time mode to facilitate the integration of real and simulated nodes. Along with network logs collected by NS3, system logs are gathered using the Linux Audit Daemon (Auditd), which leverages the Linux Auditing System for efficient log capture. These logs are processed by SPADE [19], a service that generates provenance data, enriching the dataset with an additional layer of information useful in APT detection.

The experiments were conducted in two phases. The first phase, conducted over four days, simulated normal system operations to establish a baseline behavior of the testbed components including VMs, sensors, and Raspberry Pis. The second phase, spanning three days, simulated an APT attack using APT29 tactics executed through Kali VM1 with MITRE Caldera. This phase followed the APT’s ’low and slow’ approach as attack steps are executed in random time intervals of 45 to 75 minutes to mimic the stealth and persistence typical of APTs and closely replicate real-world attack dynamics.

3.4 Dataset Properties

The dataset is organized into two folders: phase1 data and phase2 data, each containing two types of data—provenance data and network packets. The provenance data files are in CSV format and contain the nodes and edges of the provenance graph. Each node in the provenance data is assigned a unique 32-digit ID, which is utilized by the edge entries to establish connections between

Table 2. APT attack phases and techniques used in the dataset

Tactic	Technique ID	Attack Type	APT Groups
Collection	T1074	Data Staged: Local Data Staging	APT28, APT29, APT39, APT3
	T1005	Data from Local System	Andariel, APT28, APT29
	T1119	Automated Collection	APT1, APT28, Chimera
	T1113	Screen Capture	APT28, APT39, Carbanak
	T1115	Clipboard Data	APT29, APT29, APT38
Exfiltration	T1560	Archive Collected Data: Archive via Utility	APT28, APT29, APT32
	T1041	Exfiltration Over C2 Channel	Lazarus, APT3, APT32
Command and Control	T1105	Ingress Tool Transfer	Lazarus, APT29, APT3
Persistence	T1546	Event Triggered Execution	APT28, APT29, APT3
	T1136	Create Account: Local Account	Dragonfly, FIN13, APT29
Discovery	T1087	Account Discovery: Local Account	APT1, APT3, Chimera
	T1016	System Network Configuration Discovery: Internet Connection Discovery	FIN13, Gamaredon, APT29
		System Network Configuration Discovery: Wi-Fi Discovery	Magic Hound, Wizard Spider
	T1033	System Owner/User Discovery	Chimera, Dragonfly, APT3
	T1518	Software Discovery	HEXANE, MuddyWater
	T1069	Permission Groups Discovery: Local Groups	Chimera, HEXANE, APT29
	T1082	System Information Discovery	Chimera, APT3, APT32
	T1083	File and Directory Discovery	APT28, APT29, APT32
	T1018	Remote System Discovery	Chimera, APT29, APT32
Credential Access	T1552	Unsecured Credentials: Credentials In Files	APT3, APT33, FIN13
		Unsecured Credentials: Bash History	-
	T1555	Credentials from Password Stores: Credentials from Web Browsers	APT33, APT39, HEXANE
Lateral Movement	T1021	Remote Services: SSH	APT29, APT39, Lazarus
Defense Evasion	T1036	Masquerading: Right-to-Left Override	APT28, APT29, Dragonfly
	T1485	Data Destruction	APT38, Gamaredon, Lazarus

nodes in the graph.

#	**Feature**	**Description**	**Provenance Type**
1	id	Node identifier	All node types
2	type	Edge or node type	All nodes and edges
3	from	Source node ID	Edges
4	to	Destination node ID	Edges
5	uid	User Id	Process nodes
6	egid	Effective group ID	Process nodes
7	exe	Executable path	Process nodes
8	gid	Group ID	Process nodes
9	euid	Effective user ID	Process nodes
10	name	Executable name	Process nodes
11	pid	Process ID	Process nodes, WDF edges
12	seen time	Process seen time	Process nodes
13	source	Data origin	All nodes and edges
14	ppid	Parent process ID	Process nodes
15	command line	Full command line used	Process nodes
16	start time	Process start time	Process nodes
17	event ID	Unique event ID	All edges
18	time	Event time	All edges
19	operation	Type of operation	All edges
20	path	File path	File,link, directory nodes
21	subtype	Subtype of nodes	Artifact nodes
22	permissions	Access permissions	File,link, directory nodes
23	epoch	Sequence number	Artifact nodes
24	version	Version number	Artifact nodes
25	Flag	Resource access mode	Used, WGB edges
26	remote port	Port number	Network socket nodes
27	protocol	Used protocol	Network socket nodes
28	remote address	IP address	Network socket nodes
29	tgid	Thread group ID	Unknown nodes
30	fd	File descriptor	Unknown nodes
31	mode	Permission setting	WGB edges
32	label	Node label- attack/benign	All nodes
33	subLabel	Attack category	All nodes

Table 3: Provenance Data Features

Besides the IDs, the provenance data files comprise 32 features in total. However, due to the heterogeneous nature of nodes and edges that are all in a single file, not all features apply to every node or edge type, resulting in many fields being populated with NaN values. Table 3 lists all features provided in the provenance data part of the dataset. The provenance data includes two main node types: Process and Artifact. The Artifact node type is further categorized into various subtypes such as file, directory, network socket, link, and unknown, the latter being used for provenance node types that do not fit into the existing subtypes. The other data type in the dataset is the network logs captured using NS3 during the experiments and stored in pcap format. These pcap files can be further processed into CSV format. We generate the CSV format from these pcaps that have the information at the packet level and contain 67 features for each packet. The last file in the dataset is the Attack Information file, which contains all necessary information about the attacks performed during the experiments in phase 2. This information includes attack time, attack PID, and the category of attack. This file helps the researchers to further analyze the dataset behavior during the attacks.

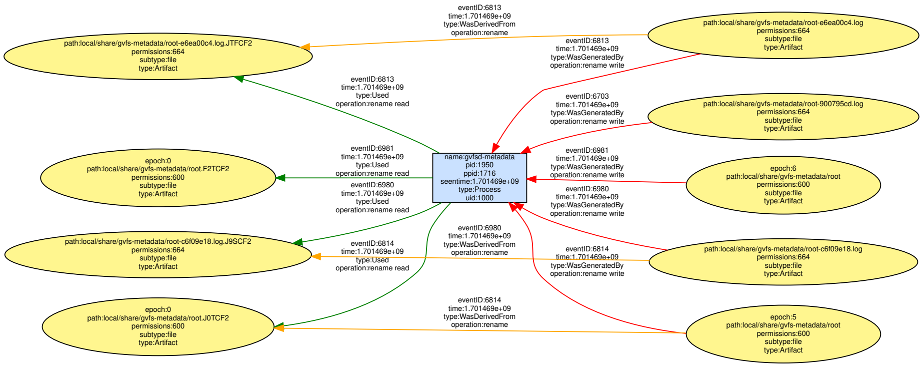

Fig. 3. A provenance graph based on a sample of our dataset

Figure 3 displays an example of a provenance graph. This is a subgraph of the complete provenance graph based on the the CICAPT-IIoT dataset. This provenance graph shows several rename operations performed on files, resulting in the generation of a new set of files.

4 Dataset Assessment

In this section, we provide an assessment of the CICAPT-IIoT dataset, evaluating its structure and utility for advancing research in APT detection within IIoT environments. We begin by presenting general statistics of the dataset to illustrate its composition and scope, followed by a comparative analysis against similar datasets. This evaluation aims to highlight the dataset's attributes and its applicability in developing robust cybersecurity solutions for the IIoT domain

4.1 General Statistics of the Dataset

The dataset was generated in two phases: Phase 1, which lasted approximately 96 hours, and Phase 2, which lasted about 72 hours. It contains of about 10 GB of data. During both phases, network logs and system logs were collected, and provenance logs were generated using the system logs and SPADE. A detailed breakdown of the dataset's segments can be found in Table 4. As shown in the Table 4, the CICAPT-IIoT dataset is notably unbalanced, with approximately 99.5% of the samples representing normal behavior and only a small fraction indicating malicious activities, which is typical in APT scenarios. This significant imbalance is reflective of real-world conditions in IIoT environments, where actual attacks are infrequent relative to regular operations.

In such contexts, using oversampling techniques to artificially balance the dataset by replicating the minority class or generating synthetic samples—can be counterproductive. Although these methods might facilitate algorithmic training in the short term, they distort the reality of how APTs manifest within network systems. By oversampling attack data, the models are trained on scenarios that are not reflective of actual operational conditions. This training approach can lead to models that perform well on balanced or altered datasets in testing environments but fail to detect genuine APT activities when deployed in real-world scenarios.

Table 4. Data distribution across different phases and data types

Attribute	**Event type**	**Provenance Data**	**Network Data**
Phase 1	**Benign**	46773 Nodes	12103705 Packets
Phase 2	**Benign**	52954 Nodes	9535819 Packets
	Attack	330	1004
	Collection	100	460
	Exfiltration	22	42
	Credential Access	82	58
	Defence Evasion	45	192
	Discovery	36	138
	Persistence	19	44
	C&C	16	56
	Lateral Movement	10	14

4.2 Comparison Against Similar Datasets

The CICAPT-IIoT dataset stands out from other datasets in several ways. First, the inclusion of multiple data sources enhances the analytical capabilities of researchers, and supports the development of new detection methods that utilize both network data and provenance logs. Furthermore, as APT attacks are known for their multi-stage operations, they require a comprehensive coverage of all associated stages, tactics, and techniques to effectively model APT campaigns in a cybersecurity dataset. Many existing datasets either do not directly address all APT tactics, as they only map network-based attacks to APT stages, or

Table 5. Related datasets analysis

Dataset	[16]	[36]	[2]	[3]	[17]	[34]	CICAPT IIoT (This Work)
IoT/IIoT	✓	✓	✓	✓			✓
Network Logs	✓	✓	✓	✓		✓	✓
Provenance/Host Logs					✓		✓
Duration	N/A	16(H)	N/A	N/A	7(D)	5(D)	7(D)
Establish foothold	✓	✓		✓		✓	✓
Collection							✓
Data exfiltration			✓		✓	✓	✓
Command & Control	✓	✓	✓		✓		✓
Persistence	✓	✓		✓	✓		✓
Discovery	✓	✓	✓	✓	✓	✓	✓
Credential Access	✓	✓	✓	✓	✓		✓
Lateral movement			✓		✓	✓	✓
Defence Evasion					✓		✓

they fail to cover all the stages necessary to fully represent an APT campaign. In contrast, the CICAPT-IIoT dataset aims to provide a complete and realistic portrayal of an APT attack, encompassing the most relevant and authentic stages and techniques. Table 5 provides an analysis of some of the related datasets. The comparison of these datasets is based on several key factors: the environment in which the data was collected, the types of data included, and the APT tactics they cover.

5 Predictive Model for APT Detection

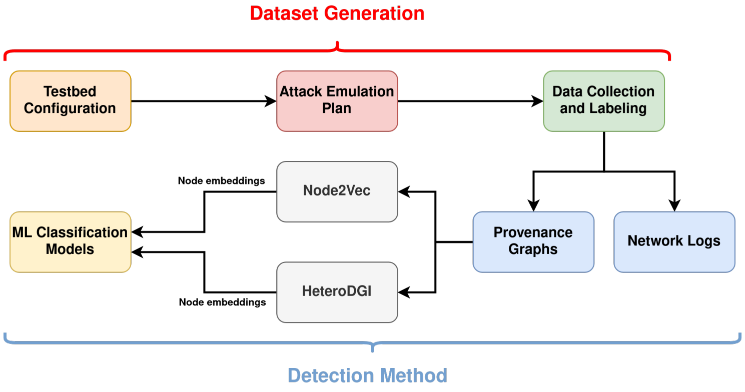

Fig. 4. Adopted approach to produce and analyze the dataset

Figure 4 shows the steps taken in this work to generate and analyze this dataset. After data collection and labeling, the provenance and network data are available to use for attack detection tasks. However, given the inherent graph structure of provenance data, the direct application of ML techniques is impractical. Therefore, an embedding method is needed to generate vector embeddings for each node in the graph. These embeddings are then utilized as inputs for various classification models to assess the dataset's effectiveness in machine learning-based APT detection tasks.

5.1 Features

Due to the heterogeneous nature of the data, encompassing various types of nodes and edges, each node type is associated with a specific subset of applicable features. The features used for each node and edge type are detailed in the table below:

Table 6. Selected features for different node and edge types

Node/Edge Type	**Attributes**
Network Socket Nodes	epoch, remote port, remote address
Process Nodes	uid, egid, exe, gid, euid, name
File Nodes	path, permissions, epoch
Directory Nodes	path, permissions
Link Nodes	path, permission, epoch
Unknown Nodes	version, tgid, fd
WTB Edges	type, operation
WGB Edges	type, operation
USED Edges	type, operation
WDF Edges	type, operation

We select these features based on the values associated with them in the dataset. Specifically, in the feature selection process, we ensure that each feature is relevant to the specific node type it describes. For instance, the "remote port" feature is used exclusively for the Network Socket node type because it is inherently related to network connections and has valid values for these nodes. Conversely, this feature is not applicable to Process nodes and therefore has NaN values for this node type. As a result, the "remote port" feature is excluded from the set of features describing Process nodes to maintain the relevance and accuracy of the feature sets for each node type. This approach ensures that the features chosen are meaningful and contribute to differentiating between nodes within the same category, which enhances the analysis of the provenance graph.

5.2 Node2Vec Based Embedding

Node2Vec [20], a popular technique, is an algorithm that generates vector representations of nodes on a graph. Using random walks, Node2Vec efficiently samples diverse neighborhoods that capture each node's essential structural properties and contextual relationships. We employ Node2Vec with a walk length of 10 to generate 64-dimensional vector representations of the nodes in the provenance graph of the dataset.

5.3 Self-Supervised Learning Based Embedding

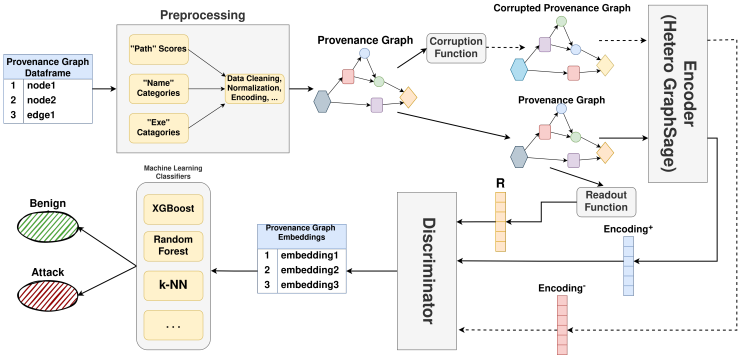

Fig. 5. Proposed self-supervised framework

Although Node2Vec embeddings offer a foundational understanding of provenance graphs, this method struggles with provenance graphs and APT detection tasks due to its inherent limitations. For instance, Node2Vec is designed for homogeneous graphs, which means, it cannot effectively handle the heterogeneity and evolving structure of provenance graphs crucial for identifying complex APT activities.

Due to these limitations, we present our Self-Supervised Learning (SSL) based model, which is specifically designed to learn node representations from provenance graphs and is based on Heterogeneous Deep Graph Infomax (HDGI) [41]. We use the Contrastive HDGI method and leverage the HeteroGraphSage as our HDGI encoder. One of the key components of the HDGI model is the encoder that is responsible for processing the graph. While many heterogeneous graph embedding methods utilize MetaPath based approaches, which rely on predefined sequences of edges, this technique is not directly applicable for provenance graphs. The dynamic nature of data interactions in provenance graphs makes it challenging to define relevant paths. Therefore, we utilize a modified GraphSAGE [21] model that is adjusted to handle heterogeneous graphs effectively. Furthermore, SSL models do not rely on labeled data, making these models suitable to adapt to the nature of APTs. Other crucial aspects of our model's design are the adaptations of the HDGI corruption function, discriminator, and readout function to suit the heterogeneous nodes of provenance graphs.

We use this SSL-based model to generate 64-sized vector embeddings for the nodes in the provenance graph. These embeddings not only reflect the node's own attributes but also embody contextual insights and relational data from its immediate environment, enhancing the overall data representation. Figure 5 presents our proposed framework for using SSL in the APT detection task.

Encoder Function GraphSage (Graph Sample and AggreGatE) [21] is a neural network model designed for graph-based data that generalizes the embedding learning process to graphs that are continuously growing. The key innovation of GraphSage is its ability to generate embeddings by sampling and aggregating features from a node's local neighborhood.

$$\mathbf{h}_{\mathcal{N}(v)}^{(k)} = \text{AGGREGATE}_k \left(\{ \mathbf{h}_u^{(k-1)} : u \in \mathcal{N}(v) \} \right) \tag{1}$$

As described in Equation 1, the features of node v's neighbors are aggregated to compute the node's new feature representation at each layer where:

- $\mathbf{h}_u^{(k-1)}$ are the features of neighbor nodes u of node v at layer $k-1$.
- $\mathcal{N}(v)$ denotes the set of neighbor nodes of v.
- AGGREGATE_k is an aggregation function, such as mean, sum, or max.

Our Heterogeneous GraphSAGE model is designed to work across diverse edge types inherent in heterogeneous graphs. The model leverages SAGEConv layers, each tailored to specific edge types as defined in the graph's metadata. This design enables it to handle the complexities and different types of edge interactions within such graphs. The model structure includes multiple layers of these convolutions, allowing for deeper feature integration across multiple hops in the graph.

Corruption Function The corruption function is crucial in self-supervised learning models like Deep Graph Infomax, primarily for generating negative samples that facilitate contrastive learning. This function alters the graph's structure and node features to create a corrupted version of the original graph, which serves as a negative sample. By differentiating between these negative samples and the original, unaltered graphs, the model learns to develop robust and generalizable node embeddings that capture the essential characteristics of the graph. This process enhances the model's ability to understand and represent the fundamental properties of the graph effectively.

Readout Function The readout function in graph neural networks is crucial for converting node-level information into a graph-level representation, which is vital for understanding the entire graph's structure, especially in applications like contrastive self-supervised learning. Standard readout methods, such as summation, which work well for homogeneous graphs, do not suit heterogeneous graphs like provenance graphs, as they tend to overlook the unique properties of different node types. To overcome this, readout functions need adaptation to

Algorithm 1 Heterogeneous graphs readout function

```
Input: embeddings - Heterogeneous graph embeddings
Output: graph_summary - Summarized embeddings
function READOUT_FUNCTION(embeddings)
    for node_types_embedding ∈ embeddings do
        types_embedding ← mean(node_types_embedding)
        graph_summary.append(types_embedding)
    end for
    return graph_summary
end function
```

handle the complexity of heterogeneous graphs. Our modified readout function processes each node type individually, ensuring that the overall graph representation maintains the distinct characteristics of each node type, thus preserving the structural and semantic integrity of heterogeneous graphs according to Algorithm 1.

6 APT Detection Model Evaluation

In this section, we analyze the provenance data component of the CICAPT-IIoT dataset to evaluate its effectiveness for provenance-based APT detection tasks. To assess the effectiveness of the provenance data component of the CICAPT-IIoT dataset in machine learning-based detection, we develop three evaluation methods. Each method utilizes node embeddings generated using the methods described in sections 5.2 and 5.3 for machine learning classification tasks. The primary goal of these classification methods is to accurately identify nodes labeled as malicious within the provenance graphs. The methods used to evaluate the dataset are as follows:

1. **Binary Classification:** Initially, the problem is defined as a binary classification task, categorizing all nodes as malicious or benign. This step aims to establish a baseline for detecting harmful entities within the system.
2. **Multi-Class Classification:** Going beyond binary classification, the analysis was expanded to include multi-class classification. This involved not only identifying benign nodes but also classifying the attack types.
3. **Attack Stages Detection:** The final experiment in the classification approach involved defining four distinct stages of attack and correlating specific attack steps to these stages, thereby creating more meaningful classes of attacks. Attack tactics were grouped based on their objectives and functionalities, resulting in this four distinct, commonly observed attack stages. Each attack stage was then treated as a separate binary classification problem.

6.1 Results and Discussion

In this section, we present the results obtained from the evaluation methods, described above and then we provide a discussion about the dataset applicability and the effectiveness of employed methods. The accuracy scores were notably

Table 7. Results of the Node2Vec-based binary classification

Model	Acc	Recall	F1
XGBoost	0.9982	0.7161	0.8270
Extra Trees	0.9981	0.7072	0.8218
k-NN	0.9980	0.7161	0.8157
Random Forest	0.9979	0.6600	0.7881
AdaBoost	0.9961	0.4741	0.6001
Decision Tree	0.9946	0.6471	0.5997
SVM	0.9943	0.2940	0.3900
Naive Bayes	0.9484	0.5482	0.1170

Table 8. Results of the Node2Vec-based multi-class classification

Model	Acc	Recall	F1
XGBoost	0.9964	0.3842	0.4012
k-NN	0.9966	0.4046	0.4182
Random Forest	0.9967	0.3913	0.4242
AdaBoost	0.9934	0.1820	0.1954
SVM	0.9969	0.3876	0.4151
Extra Trees	0.9967	0.3913	0.4170
Decision Tree	0.9935	0.3735	0.3553
Naive Bayes	0.8203	0.4564	0.2546

Table 9. Models performance metrics for Node2Vec-based attack stage detection

Attack Stage	Model	Accuracy	Recall	Precision	F1-Score
Collection and Exfiltration	XGBoost	0.9991	0.6278	0.9732	0.7446
	Random Forest	0.9988	0.5069	0.9750	0.6546
	AdaBoost	0.9987	0.5306	0.8714	0.6451
Credential Access and C&C	XGBoost	0.9993	0.6405	1.0000	0.7719
	Random Forest	0.9993	0.6403	1.0000	0.7698
	AdaBoost	0.9994	0.7143	0.9333	0.8031
Defense Evasion and Persistence	XGBoost	0.9993	0.4500	1.000	0.5931
	Random Forest	0.9990	0.1900	0.6000	0.2800
	AdaBoost	0.9992	0.4100	0.8167	0.5098
Discovery and Lateral Movement	XGBoost	0.9995	0.4083	0.8000	0.5406
	Random Forest	0.9993	0.1500	0.4000	0.2100
	AdaBoost	0.9994	0.4000	0.8083	0.5357

high for all ML models in every classification tasks. However, due to the significant class imbalance in the dataset—approximately 99.5% of instances are labeled as benign, the accuracy alone may not be informative enough. As a result, our evaluation puts more weight on criteria like Recall and the F1 score. Table 7 presents the performance metrics of binary classification models that utilize embeddings generated by the Node2Vec method. While the results display high overall accuracy for all the models, the average recall score is around 70%. The good performance of these models, despite utilizing an embedding method like Node2Vec, underscores the applicability of the CICAPT-IIoT dataset for graph-based APT detection tasks. The overall performance of all models decreases in the attack classification task, as shown in Table 8, due to the increased complexity of this classification challenge. These results suggest that while it is possible to detect malicious nodes in the provenance graph using simpler embedding methods, accurately classifying the attack categories proves more challenging and requires more sophisticated approaches.

The results of the Node2Vec-based attack stage detection are shown in Table 9. All stages are detectable by machine learning models with an average recall score of approximately 60%. This demonstrates that defining distinct attack

Table 10. Results of the SSL-based binary class classification

Model	Acc	Recall	F1
Extra Trees	0.9986	0.8444	0.8785
Random Forest	0.9986	0.8232	0.8760
XGBoost	0.9984	0.8359	0.8638
Decision Tree	0.9983	0.8361	0.8615
AdaBoost	0.9980	0.7281	0.8153
k-NN	0.9978	0.7629	0.8097
Naive Bayes	0.5143	0.9699	0.0242

Table 11. Results of the SSL-based multi-class classification

Model	Acc	Recall	F1
XGBoost	0.9975	0.5816	0.6277
k-NN	0.9971	0.4340	0.4724
Random Forest	0.9975	0.5789	0.6299
AdaBoost	0.9934	0.1850	0.1670
Extra Trees	0.9973	0.5657	0.6032
Decision Tree	0.9967	0.5471	0.5264
Naive Bayes	0.4284	0.3139	0.2338

stages can be effective for detecting APT attacks and providing a holistic view of the APT campaign. However, improvements may be necessary, suggesting the need for more advanced approaches to enhance performance.

These results from Node2Vec embeddings, demonstrate the potential of machine learning methods to classify APTs within IIoT operations. Indeed, this serves as a baseline for any machine learning-based APT detection task.

To employ an embedding method better suited to the features of our dataset, we utilize our proposed SSL model described in Section 5.3, applying the same evaluation methods for consistency in our analysis. Table 10 shows the results of the binary classification task using the SSL method. All models show improved performance compared to the Node2Vec model, with recall scores approximately 10% higher. These improvements suggest that SSL-based detection methods have a better capability of identifying malicious nodes within the provenance graph. These improvements are also noticeable in Table 12, where we utilized SSL-based embeddings for the attack stage detection task. Here, recall and F1 scores have shown a significant boost, further validating the effectiveness of SSL methods in more complex classification scenarios.

The performance of the SSL model compared to the baseline approach emphasizes that APT detection tasks using provenance graphs require methods tailored to the unique characteristics of such attacks and their data types. Figure 6 shows the F1-score comparison of the Node2Vec and SSL-based approach in the attack stage detection task.

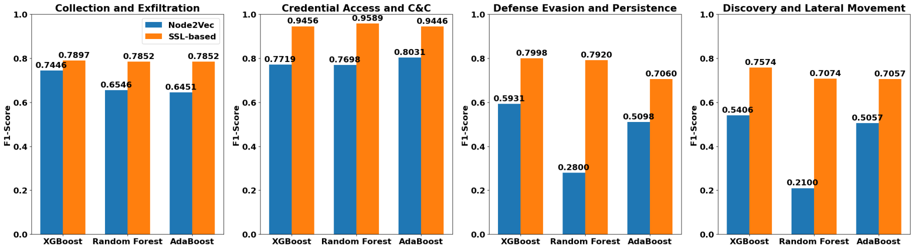

Fig. 6. Attack stage detection comparison

Table 12. Models performance metrics for SSL-based attack stage detection

Attack Stage	Model	Accuracy	Recall	Precision	F1-Score
Collection and Exfiltration	XGBoost	0.9992	0.7294	0.8881	0.7897
	Random Forest	0.9991	0.7292	0.8784	0.7852
	AdaBoost	0.9991	0.7292	0.8784	0.7852
Credential Access and C&C	XGBoost	0.9999	0.9262	0.9732	0.9456
	Random Forest	0.9999	0.9262	1.0000	0.9589
	AdaBoost	0.9998	0.9262	0.9708	0.9446
Defense Evasion and Persistence	XGBoost	0.9996	0.7300	0.9300	0.7998
	Random Forest	0.9996	0.7100	0.9417	0.7920
	AdaBoost	0.9993	0.6850	0.7693	0.7060
Discovery and Lateral Movement	XGBoost	0.9997	0.6833	0.9417	0.7574
	Random Forest	0.9997	0.6500	0.8417	0.7074
	AdaBoost	0.9996	0.6833	0.7750	0.7057

7 Conclusion

Given the escalating threat of APT attacks on IIoT systems, developing effective detection solutions is crucial. Datasets are central to these efforts as they enable the development of defenses against such sophisticated threats. In this paper, we present CICAPT-IIoT, a dataset designed for IIoT environments, aimed at helping researchers in security analysis and the design of detection techniques against APTs. The dataset contains over 20 well-known attack techniques, forming 8 different tactics commonly utilized in APT campaigns. The collected data in provenance and network log formats are available in the CIC website[4]. Furthermore, we provide a thorough analysis of the dataset using well-known machine learning models to aid the researchers in developing more effective methods. Finally, we use the CICAPT-IIoT dataset to propose a self-supervised learning-based method for the APT detection task.

Several areas remain open for exploration in our future work. First, our dataset can be utilized to develop and refine real-time detection algorithms capable of identifying threats as they unfold. Additionally, incorporating more diverse IIoT devices and attack scenarios would further improve the generalizability of detection models.

Acknowledgments. The authors graciously acknowledge the support from the Canadian Institute for Cybersecurity (CIC), the funding support from the National Research Council of Canada (NRC) through the AI for Logistics collaborative program, the NSERC Discovery Grant (no. RGPIN 231074), and Tier 1 Canada Research Chair to Dr. Ghorbani.

Disclosure of Interests. The authors declare no conflict of interest.

[4] CIC website: `https://www.unb.ca/cic/datasets/iiot-dataset-2024.html`

References

1. Al-Hawawreh, M., Sitnikova, E.: Developing a security testbed for industrial internet of things. IEEE Internet of Things Journal **8**(7), 5558–5573 (2020)
2. Al-Hawawreh, M., Sitnikova, E., Aboutorab, N.: X-iiotid: A connectivity-agnostic and device-agnostic intrusion data set for industrial internet of things. IEEE Internet of Things Journal **9**(5), 3962–3977 (2021)
3. Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., Anwar, A.: Ton_iot telemetry dataset: A new generation dataset of iot and iiot for data-driven intrusion detection systems. Ieee Access **8**, 165130–165150 (2020)
4. Alshamrani, A., Myneni, S., Chowdhary, A., Huang, D.: A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities. IEEE Communications Surveys & Tutorials **21**(2), 1851–1877 (2019)
5. Aman, M.N., Basheer, M.H., Sikdar, B.: Data provenance for iot with light weight authentication and privacy preservation. IEEE Internet of Things Journal **6**(6), 10441–10457 (2019)
6. Aman, M.N., Chua, K.C., Sikdar, B.: Secure data provenance for the internet of things. In: Proceedings of the 3rd ACM international workshop on IoT privacy, trust, and security. pp. 11–14 (2017)
7. Anjum, M.M., Iqbal, S., Hamelin, B.: Anubis: a provenance graph-based framework for advanced persistent threat detection. In: Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing. pp. 1684–1693 (2022)
8. Barre, M., Gehani, A., Yegneswaran, V.: Mining data provenance to detect advanced persistent threats. In: 11th International Workshop on Theory and Practice of Provenance (TaPP 2019) (2019)
9. Bates, A., Tian, D.J., Butler, K.R., Moyer, T.: Trustworthy {Whole-System} provenance for the linux kernel. In: 24th USENIX Security Symposium (USENIX Security 15). pp. 319–334 (2015)
10. Berrada, G., Cheney, J.: Aggregating unsupervised provenance anomaly detectors. In: 11th International Workshop on Theory and Practice of Provenance (TaPP 2019) (2019)
11. Breitenbacher, D., Homoliak, I., Aung, Y.L., Tippenhauer, N.O., Elovici, Y.: Hades-iot: A practical host-based anomaly detection system for iot devices. In: Proceedings of the 2019 ACM Asia conference on computer and communications security. pp. 479–484 (2019)
12. Chen, Z., Liu, J., Shen, Y., Simsek, M., Kantarci, B., Mouftah, H.T., Djukic, P.: Machine learning-enabled iot security: Open issues and challenges under advanced persistent threats. ACM Computing Surveys **55**(5), 1–37 (2022)
13. Corporation, M.: Apt29. `https://attack.mitre.org/groups/G0016/` (2023), accessed: Oct-2023
14. Da Xu, L., He, W., Li, S.: Internet of things in industries: A survey. IEEE Transactions on industrial informatics **10**(4), 2233–2243 (2014)
15. Di Pinto, A., Dragoni, Y., Carcano, A.: Triton: The first ics cyber attack on safety instrument systems. Proc. Black Hat USA **2018**, 1–26 (2018)
16. Ferrag, M.A., Friha, O., Hamouda, D., Maglaras, L., Janicke, H.: Edge-iiotset: A new comprehensive realistic cyber security dataset of iot and iiot applications for centralized and federated learning. IEEE Access **10**, 40281–40306 (2022)
17. Five Directions: Operationally transparent cyber (optc) dataset, `https://github.com/FiveDirections/OpTC-data`, accessed: 2024-02-22

18. Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., Vázquez, E.: Anomaly-based network intrusion detection: Techniques, systems and challenges. computers & security **28**(1-2), 18–28 (2009)
19. Gehani, A., Tariq, D.: Spade: Support for provenance auditing in distributed environments. In: ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing. pp. 101–120. Springer (2012)
20. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 855–864 (2016)
21. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. Advances in neural information processing systems **30** (2017)
22. Han, X., Pasquier, T., Bates, A., Mickens, J., Seltzer, M.: Unicorn: Runtime provenance-based detector for advanced persistent threats. arXiv preprint arXiv:2001.01525 (2020)
23. Hassan, W.U., Bates, A., Marino, D.: Tactical provenance analysis for endpoint detection and response systems. In: 2020 IEEE Symposium on Security and Privacy (SP). pp. 1172–1189. IEEE (2020)
24. Herschel, M., Diestelkämper, R., Ben Lahmar, H.: A survey on provenance: What for? what form? what from? The VLDB Journal **26**, 881–906 (2017)
25. Hossain, M.N., Milajerdi, S.M., Wang, J., Eshete, B., Gjomemo, R., Sekar, R., Stoller, S., Venkatakrishnan, V.: {SLEUTH}: Real-time attack scenario reconstruction from {COTS} audit data. In: 26th USENIX Security Symposium (USENIX Security 17). pp. 487–504 (2017)
26. Hu, R., Yan, Z., Ding, W., Yang, L.T.: A survey on data provenance in iot. World Wide Web **23**, 1441–1463 (2020)
27. Jaloudi, S.: Communication protocols of an industrial internet of things environment: A comparative study. Future Internet **11**(3), 66 (2019)
28. Lahmar, H.M.D.R.B.: H a survey on provenance: What for? what form? what from. VLDB J **26**(6), 881 (2017)
29. Langner, R.: Stuxnet: Dissecting a cyberwarfare weapon. IEEE Security & Privacy **9**(3), 49–51 (2011)
30. Malik, P.K., Sharma, R., Singh, R., Gehlot, A., Satapathy, S.C., Alnumay, W.S., Pelusi, D., Ghosh, U., Nayak, J.: Industrial internet of things and its applications in industry 4.0: State of the art. Computer Communications **166**, 125–139 (2021)
31. Michael, Z., Florian, G., Elizabeth, C., Tharam, D.: Provenance-based intrusion detection systems: a survey. ACM Comput. Surv. **55**, 36 (2022)
32. Milajerdi, S.M., Gjomemo, R., Eshete, B., Sekar, R., Venkatakrishnan, V.: Holmes: real-time apt detection through correlation of suspicious information flows. In: 2019 IEEE Symposium on Security and Privacy (SP). pp. 1137–1152. IEEE (2019)
33. MITRE: Group g0016 - APT29. `https://attack.mitre.org/groups/G0016/`, accessed: April, 2024
34. Myneni, S., Chowdhary, A., Sabur, A., Sengupta, S., Agrawal, G., Huang, D., Kang, M.: Dapt 2020-constructing a benchmark dataset for advanced persistent threats. In: Deployable Machine Learning for Security Defense: First International Workshop, MLHat 2020, San Diego, CA, USA, August 24, 2020, Proceedings 1. pp. 138–163. Springer (2020)
35. Myneni, S., Jha, K., Sabur, A., Agrawal, G., Deng, Y., Chowdhary, A., Huang, D.: Unraveled—a semi-synthetic dataset for advanced persistent threats. Computer Networks **227**, 109688 (2023)

36. Neto, E.C.P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R., Ghorbani, A.A.: Ciciot2023: A real-time dataset and benchmark for large-scale attacks in iot environment. Sensors **23**(13), 5941 (2023)
37. NS-3: Ns-3 consortium, `https://www.nsnam.org/`, accessed: 2024-03-11
38. Nwafor, E., Campbell, A., Hill, D., Bloom, G.: Towards a provenance collection framework for internet of things devices. In: 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI). pp. 1–6. IEEE (2017)
39. Pan, B., Stakhanova, N., Ray, S.: Data provenance in security and privacy. ACM Computing Surveys (2023)
40. Pasquier, T., Han, X., Goldstein, M., Moyer, T., Eyers, D., Seltzer, M., Bacon, J.: Practical whole-system provenance capture. In: Proceedings of the 2017 Symposium on Cloud Computing. pp. 405–418 (2017)
41. Ren, Y., Liu, B., Huang, C., Dai, P., Bo, L., Zhang, J.: Heterogeneous deep graph infomax. arXiv preprint arXiv:1911.08538 (2019)
42. Sadineni, L., Pilli, E.S., Battula, R.B.: Provnet-iot: Provenance based network layer forensics in internet of things. Forensic Science International: Digital Investigation **43**, 301441 (2022)
43. Sisinni, E., Saifullah, A., Han, S., Jennehag, U., Gidlund, M.: Industrial internet of things: Challenges, opportunities, and directions. IEEE transactions on industrial informatics **14**(11), 4724–4734 (2018)
44. Stojanović, B., Hofer-Schmitz, K., Kleb, U.: Apt datasets and attack modeling for automated detection methods: A review. Computers & Security **92**, 101734 (2020)
45. Whitehead, D.E., Owens, K., Gammel, D., Smith, J.: Ukraine cyber-induced power outage: Analysis and practical mitigation strategies. In: 2017 70th Annual Conference for Protective Relay Engineers (CPRE). pp. 1–8. IEEE (2017)
46. Wurm, J., Hoang, K., Arias, O., Sadeghi, A.R., Jin, Y.: Security analysis on consumer and industrial iot devices. In: 2016 21st Asia and South Pacific design automation conference (ASP-DAC). pp. 519–524. IEEE (2016)
47. Xu, K., Tian, K., Yao, D., Ryder, B.G.: A sharper sense of self: Probabilistic reasoning of program behaviors for anomaly detection with context sensitivity. In: 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). pp. 467–478. IEEE (2016)
48. Yang, Z., Liu, X., Li, T., Wu, D., Wang, J., Zhao, Y., Han, H.: A systematic literature review of methods and datasets for anomaly-based network intrusion detection. Computers & Security **116**, 102675 (2022)
49. Zipperle, M., Gottwalt, F., Chang, E., Dillon, T.: Provenance-based intrusion detection systems: A survey. ACM Computing Surveys **55**(7), 1–36 (2022)