A Survey on Threat Hunting in Enterprise Networks

Boubakr Nour, Makan Pourzandi, and Mourad Debbabi

***Abstract*—With the rapidly evolving technological landscape, the huge development of the Internet of Things, and the embracing of digital transformation, the world is witnessing an explosion in data generation and a rapid evolution of new applications that lead to new, wider, and more sophisticated threats that are complex and hard to be detected. Advanced persistence threats use continuous, clandestine, and sophisticated techniques to gain access to a system and remain hidden for a prolonged period of time, with potentially destructive consequences. Those stealthy attacks are often not detectable by advanced intrusion detection systems (*e.g.,* LightBasin attack was detected in 2022 and has been active since 2016). Indeed, threat actors are able to quickly and intelligently alter their tactics to avoid being detected by security defense lines (*e.g.,* prevention and detection mechanisms). In response to these evolving threats, organizations need to adopt new proactive defense approaches. Threat hunting is a proactive security line exercised to uncover stealthy attacks, malicious activities, and suspicious entities that could circumvent standard detection mechanisms. Additionally, threat hunting is an iterative approach to generate and revise threat hypotheses endeavoring to provide early attack detection in a proactive way. The proactiveness consists of testing and validating the initial hypothesis using various manual and automated tools/techniques with the objective of confirming/refuting the existence of an attack. This survey studies the threat hunting concept and provides a comprehensive review of the existing solutions for Enterprise networks. In particular, we provide a threat hunting taxonomy based on the used technique and a sub-classification based on the detailed approach. Furthermore, we discuss the existing standardization efforts. Finally, we provide a qualitative discussion on current advances and identify various research gaps and challenges that may be considered by the research community to design concrete and efficient threat hunting solutions.**

***Index Terms*—Cybersecurity, cyber threat intelligence, threat hunting, threat detection.**

I. Introduction

THE SCIENTIFIC arms race and the recent pandemic the world witnessed have brought the threat of cyber-attacks on individuals, businesses, and nations into sharp focus. To cope with the lockdowns, most businesses have adopted online/hybrid work environments that created large opportunities for attackers and hackers to strike. Today's warfare is fought in both digital and actual worlds, which is enhanced by state-sponsored threat actors who are striving to undermine essential infrastructure. According to the State of Cybersecurity Resilience report [1], the security attacks increased by 31% from 2020 to 2021. Cisco/Cybersecurity Ventures [2] foreshadows that the cost of cybercrime will hit $10.5 trillion by 2025.

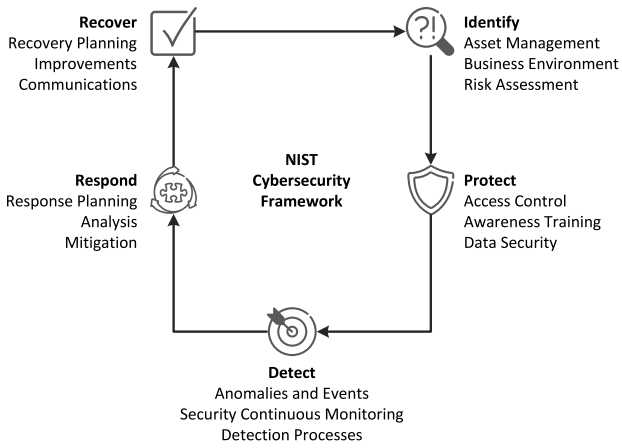

Fig. 1. NIST cybersecurity framework [3].

In order to alleviate the risk and develop an overarching understanding of security, the National Institute of Standards and Technology (NIST) introduced a Cybersecurity Framework [3] (illustrated in Figure 1) that provides a policy framework, which can be used by organizations to assess and improve their ability to prevent, detect, and respond to cyber-attacks. The core of the framework reflects a continuous process including the function areas: (i) *Identify:* develop the organizational understanding to manage the cybersecurity risk of systems, assets, data, and capabilities; (ii) *Protect:* develop and implement appropriate safeguards to ensure the delivery of critical services; (iii) *Detect:* develop and implement appropriate activities to identify the occurrence of a cybersecurity event; (iv) *Respond:* develop and implement appropriate activities to take action regarding a detected cybersecurity incident; (v) *Recover:* develop and implement appropriate activities to maintain plans for resilience and to restore any capabilities or services that were impaired due to a cybersecurity incident. Toward this, Security Operations Center (SOC) deploys multiple approaches to (i) minimize attacks using detection and prevention techniques [4] and (ii) understand attacks using forensic analysis approaches [5]. Both procedures require

Manuscript received 30 November 2022; revised 24 April 2023 and 2 June 2023; accepted 25 July 2023. Date of publication 14 August 2023; date of current version 22 November 2023. This work was supported in part by Ericsson Research; in part by Concordia University; and in part by the National Cybersecurity Consortium (NCC) under the Cyber Security Innovation Network (CSIN). *(Corresponding author: Boubakr Nour.)*

Boubakr Nour was with the Gina Cody School of Engineering and Computer Science, Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC H3G 1M8, Canada. He is now with the GFTL Security Research, Ericsson, Montreal, QC H4S 0B6, Canada (e-mail: boubakr.nour@ericsson.com).

Makan Pourzandi is with the GFTL Security Research, Ericsson, Montreal, QC H4S 0B6, Canada (e-mail: makan.pourzandi@ericsson.com).

Mourad Debbabi is with the Gina Cody School of Engineering and Computer Science, Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC H3G 1M8, Canada (e-mail: mourad.debbabi@concordia.ca).

Digital Object Identifier 10.1109/COMST.2023.3299519

diverse data collection and audit logs from multiple hosts, applications, and network interfaces for analytical purposes and ultimately decision-making [6], [7], [8].

Contrary to threat detection, threat hunting [9] is an iterative approach to monitoring, generating, and revising threat hypotheses. The objective of threat hunting is to actively detect attacks, catch suspicious activities, and apprehend malicious nodes at a primeval stage in a complex network environment, *e.g.,* enterprise or Telecom network [10]. By actively, we mean that we could presume the existence of a threat actor based on some activities (*i.e.,* hypothesis), initiate actions to test the hypothesis, *e.g.,* instantiate targeted collector agents in defined entities, collect required data, and then validate the initial hypothesis to uncover new threats/attacks [11], [12].

Commercialized security solutions (*e.g.,* Security Information and Event Management (SIEM), Security Orchestration, Automation, and Response (SOAR), eXtended Security Orchestration, Automation, and Response (XSOAR)) have a variety of security capabilities such as vulnerability monitoring and assessment, event analysis, and digital forensics. Yet, it is essential to extend them with threat hunting capabilities in order to strengthen the security posture of enterprises and boost investigative efficiency [13], [14], [15], [16]. It is worth mentioning that the designed threat hunting architecture is not an incident-response process. This is due to the fact that the incident-response model [17] is reactive in nature and deals with incidents after they occur, while threat hunting [18] is a proactive model that seeks to identify threats before they happen and cause damage. More specifically, an incident-response approach refers to a set of procedures taken by a security team after a security incident (*e.g.,* an attack) has occurred. Its goal is to manage the situation aiming at minimizing damage and reducing the recovery time and costs. An incident-response plan often includes a set of instructions for detection, forensics, investigation, and recovery from security incidents [19]. Threat hunting, on the contrary, is a proactive security process where security analysts actively scrutinize system telemetry based on their expertise to identify threats that may not have been detected by conventional security solutions. This process involves generating threat hypotheses on potential attacks based on the understanding of the system and the threat landscape, then investigating those hypotheses with the aim of discovering new threats [20].

In fact, a lot of efforts and solutions have been presented in both academia and industry sectors to enhance Enterprise cybersecurity [21], [22], [23], [24], such as detection [25], prevention [26], and mitigation [27]. However, threat hunting has not been well covered. To the best of our knowledge, this is the first survey focusing on threat hunting in Enterprise networks. According to studies conducted in [28], [29], [30], the concept of threat hunting is still unexplored and poorly defined from both procedural and organizational standpoints. Indeed, most organizations are still reacting to alerts and incidents instead of proactively seeking out threats. Studies also uncover that intriguing observations such as threat intelligence and hunting must go hand in hand to perform effectively and need to be automated. The same studies also show that organizations that achieve measurable improvements in their security, most measure improvements in speed and accuracy, where the use of hunting lessens their exposures.

The existing work lacks an in-depth study of threat hunting. Current work either focuses on particular security aspects (*e.g.,* threat modeling, threat detection, incident response) or completely ignores threat hunting. To fill the gaps, we comprehensively investigate threat hunting by covering its methodology and state-of-the-art efforts. It is worth mentioning that, in this work, we primarily focus on threat hunting and not threat detection. In addition, we narrow down the threat hunting application domain to Enterprise networks and not malware (*e.g.,* viruses) [31], [32], [33], [34] or operating systems (*e.g.,* kernel) [35], [36], [37], [38]. We also discuss the existing standardization efforts and open-source solutions, as well as highlight issues and open research challenges with open-ended discussion on the successful proliferation of threat hunting in Enterprise networks. The contributions of this work are summarized below:

- We comprehensively survey threat hunting by covering its models, methodology, process, and components.
- We design a unified top-down architecture for threat hunting. We define different components, how they interact with each other in a closed control loop, and their input and outputs. To the best of our knowledge, this is the first systematic detailed component-based architecture of threat hunting.
- We describe in detail, the current threat hunting state-of-the-art solutions in Enterprise networks based on used techniques such as machine learning, graph-based solutions, rule-based, and statistical-based methods. We further classify these solutions based on the detailed approach.
- We provide an overview of the existing standardization and open-source efforts, including representation, query languages, operational, sharing platforms, and testbeds.
- We carry out a gap analysis and discussion for the reviewed solutions along with some recommendations to amend the identified gaps.
- We outline several challenges faced by threat hunting and highlight future research directions.

The rest of the survey is organized as follows. The next section reviews the related work and surveys and highlights our major contribution. In Section III, we introduce the concept of threat hunting including the threat actors, hunting methodology, types, and hunting maturity model. Section IV presents our top-down threat hunting general architecture and its modules and components. A detailed overview of the existing threat hunting solutions is presented in Section V. These solutions are grouped based on the used technique and then sub-classified based on the detailed approach. In Section VI, we present the threat hunting efforts in standardization bodies, open source, and industry efforts. Section VII provides a summary discussion of the conducted study. In Section VIII, we focus on the lessons learned, mainly on limitations, issues, and challenges of threat hunting. We also highlight some future research directions. Finally, we conclude the paper in Section IX.

TABLE I
COMPARISON OF OUR SURVEY CONTRIBUTIONS TO THE RELATED SURVEY PAPERS

Ref.	Objectives	Year	Covered Solutions	Year Interval	Threat Hunting	Scope
[39]	✓ Overview on security threat modeling ✓ Review threat modeling solutions	2019	54	2004 - 2017	❍	✗ Limited to threat modeling ✗ Missing threat hunting concepts
[40]	✓ Focus on cross-architectural IoT malware detection and classification ✓ Cover ML detection techniques	2021	28	2015 - 2021	◗	✗ Limited overview on threat hunting ✗ More focus on threat detection rather than hunting
[17]	✓ Overview core concepts required for incident response ✓ Focus on standardization approaches for incident response	2021	/	/	❍	✗ Limited to incident response ✗ Missing academia efforts ✗ Missing threat hunting concept
[41]	✓ Focus on graphical security modeling ✓ Cover attack tree solutions	2019	13	2012 - 2018	❍	✗ Limited to attack tree–based modeling ✗ Missing threat hunting concept
[42]	✓ Overview on provenance graph concept ✓ Review of provenance graph-based threat detection and investigation	2021	38	2012 - 2020	◗	✗ Limited to threat detection ✗ Limited to provenance graph representation
[43]	✓ Brief study on threat hunting ✓ Focus on SideWinder APT use case	2022	8	2015-2020	❍	✗ Very limited number of reviewed papers ✗ Non-comprehensive survey and methodology
[44]	✓ Focus on digital forensics ✓ Review of IoT forensics solutions	2020	58	2010 - 2018	❍	✗ Limited to digital forensics in IoT
[45]	✓ Commercial and technical analysis of SIEM solution	2021	/	/	❍	✗ Limited to SIEM solutions ✗ Not considering threat hunting features
Our	✓ Overview on threat hunting concepts and general architecture ✓ Cover a large number of recent work categorized into new taxonomy ✓ Comprehensive review of state-of-the-art papers ✓ Cover standardization efforts ✓ Highlight research directions	2023	39	2017-2023	●	/

❍: No literature review, ◗: Partial literature review, ●: Comprehensive literature review.

II. RELATED WORK AND SURVEYS

Existing literature focuses on security aspects such as threat classification [46] and cyber forensics [47]. These solutions mainly address the reactive investigation rather than providing a proactive defense line. A handful of research and surveys are available in the literature that focuses on threat modeling [39], [48], [49], [50], threat detection [40], [51], [52], [53], and incident response [17], [54], [55]. However, the current literature does not cover threat hunting, which is an essential security aspect in today's SOC. To fill the gap, we carry out a comprehensive and systematic survey on threat hunting for Enterprise networks. To justify the need and contribution of this work, we summarize the existing surveys in Table I and the details are as follows.

Xiong and Lagerström [39] carried out a systematic literature review on threat modeling based on systematic queries. The authors covered efforts that (i) contribute to the threat modeling, (ii) use threat modeling approach, and (iii) enhance the threat modeling process. Based on the study, the authors underlined some challenges and research directions that could be addressed by researchers such as automation, validation methods, and integration with vulnerability databases. Raju et al. [40] surveyed the latest developments in research methods for Internet of Things (IoT) malware threat detection and hunting. The authors categorized the existing solutions based on learning malware features and analyzed their usability for detecting cross-architectural threats. Some research challenges and directions have been highlighted. Schlette et al. [17] examined the existing standardization approaches and efforts for security incident response. The paper provides a comprehensive analysis of incident response formats (*i.e.,* frameworks, standards, scoring systems, enumerations, and serializations). Yet, the authors did not cover academic solutions nor threat-hunting efforts. Wideł et al. [41] conducted a research survey on attack tree-based modeling with a focus on the semantic, generation, and qualitative approaches. The authors mainly considered attack trees and attack-defense trees. A formal method and graphical security modeling have been presented along with their application, interpretation, semi-automated creation, and quantitative analysis of attack trees. Some research challenges have also been highlighted. Similarly, Li et al. [42] focused on threat detection and investigation utilizing a system-level provenance graph. The authors classified the existing provenance graph-based threat detection solutions into three main categories, namely, data collection, management, and detection. In addition, each category has sub-classes based on the applied technique and algorithm. Some insights and challenges have also been presented to guide future research in this area. Chen et al. [43] briefly studied threat hunting by focusing mainly on SideWinder APT conducted against Pakistani government organizations. The authors presented the APT attack and then review a few machine-learning solutions that might help in APT hunting. Hou et al. [44] investigated the impact of

IoT on digital forensics and conducted a survey on the existing research efforts. The latter has been categorized into three dimensions, namely, temporal, spatial, and technical dimensions. Each dimension has different classes based on the applied technique and the targeted aspect. The authors highlighted some open issues and provided research directions in order to advance IoT digital forensics.

From the aforementioned discussion and Table I, it is important to note that the existing surveys concentrate on a specific security aspect (*e.g.,* threat modeling, incident response, threat detection) while neglecting the threat discovering and hunting. Furthermore, the research carried out in [17], [39], [40], [41], [44] does not cover threat hunting. Whereas, work in [42] is the closest effort to our work, yet it focuses more on threat detection rather than hunting. Table I also provides a comparative summary of the existing surveys and how they differ from our survey including their covered solutions, their respective year interval, and limitations. To the best of our knowledge, this work is the first comprehensive effort to review threat hunting solutions and design a unified top-down architecture for threat hunting. Our contribution can be viewed as a critical and systematic overview of threat hunting along with state-of-the-art solutions, issues, gaps, and future challenges.

III. Threat Hunting: A Bird's-Eye View

Threat hunting tends to uncover any security weaknesses in the network, ranging from suspicious activities to large-scale APT campaigns [56] that could not be detected by conventional threat detection solutions. The process may be triggered by observing indicators or evidence that might lead to a threat or the identification of new threats [42] (see Section III-B for details). Afterward, the security analysts investigate the system telemetry and log files among others using various security tools and analysis techniques to validate or refute the existence of threats in the system [57]. The main task of threat hunting is to "interactively program human reasoning procedures, inspect the outcomes, and iteratively revise the threat hypotheses and the reasoning procedures based on observations and related knowledge" [9]. This section provides an overview of threat hunting. We start by describing different threat actors, and then we introduce existing threat hunting methodologies, types, and maturity models.

A. Threat Actors

Every enterprise is exposed to various threats that need to be hunted and detected at an early stage [58]. Threat actors can broadly be categorized, based on the actors, into two main classes [59]:

1) *Insider Threat Actors:* The U.S. Cybersecurity and Infrastructure Security Agency (CISA) [60] defines insiders as threat actors connected to (or inside) the enterprise and have privileged access to target the system [61]. Those actors use their authorized access and can harm the network's resources and perform malicious activities either unintentionally or intentionally [62]. Insider threat actors are challenging to be detected due to the following reasons [63]: (i) *Legitimate access*: since insiders have legitimate access to systems, it is difficult to distinguish their malicious activities from regular actions; (ii) *Knowledge and evasion*: as insiders are often familiar with an organization's security protocols and they have enough knowledge about the organization's policies, it is easier for them to evade detection; and (iii) *Trust and privileges*: due to the fact that insiders are often trusted members of an organization, it makes it harder for security systems to flag their activities as suspicious, which allows them to elevate privileges and bypass certain security measures. Threat hunting for the insiders becomes more difficult as their actions being considered as legitimate are more difficult to raise any alarm to trigger threat hunting. Furthermore, evading the tracing and obfuscating of their actions make it more difficult to track their actions in order to form a valid hypothesis or test the formed hypothesis. Insider actors can further be divided into (i) *authorized insiders*: actors with legitimate access to the system, and (ii) *opportunistic insiders*: actors without legitimate access who take opportunities to access and damage the system [64]. An authorized insider can be a (disgruntled) current employee, while an opportunistic insider can be a former employee (with unrevoked access) or any negligent third-party contractor.
2) *Outsider Threat Actors:* Outsiders are threat actors that are not authorized to connect to the enterprise but attempt to access, manipulate, interrupt, or destroy the enterprise network/data [65]. While outsider threats can be challenging to detect, they are often more visible compared to insider threats due to the following reasons [63]: (i) *Network monitoring*: employing network monitoring systems can help in tracking and detecting suspicious external activities by flagging unusual traffic patterns, unauthorized access attempts, or known attack signatures; (ii) *Threat intelligence*: having access to threat intelligence feeds can help in providing information about known malicious actors and their techniques, which in return help in identifying outsider threats; and (iii) *Attack signatures*: relying on known attack signatures, such as malware or specific patterns of network traffic can assist the security team to detect, block, or mitigate outsider threats. Therefore, the threat hunting for outsiders becomes less difficult as their access could be better monitored and traced. In addition, there is much more information about the external actors and their modus operandi/attack signatures, making the hypothesis generation more precise and more efficient. Based on the intention, outsider threat actors can be grouped into four main groups:
   - *Nation-states:* well-trained actors (*e.g.,* government intelligence agencies) with extensive experience and expertise to perform advanced and stealthy attacks. Their objective is to gain intelligence of national interest or to disrupt infrastructures of strategic importance.

- *Cybercriminals:* individuals that exploit technology to intentionally damage/steal/eavesdrop on an enterprise for personal gains/profits.
- *Hacktivists:* individuals or groups motivated to hack with political or social causes. They may target government agencies, corporations, or other organizations that they feel are acting against their beliefs.
- *Competitors:* any type of competitor in the same enterprise's market that aim to gain a competitive advantage.

Overall, both insider and outsider threat actors pose risks to organizations [66], [67]. The threat actor characteristics widely change given the context (*i.e.,* utilities in a geo-political area) and asset value (*e.g.,* banking, social media). Thus, the threat hunting process should take into account the threat actor characteristics, including: (i) the intent, *e.g.,* stealing services by the criminals, eavesdropping by competitors, destruction and DoS by hacktivists; (ii) the skill sets, *e.g.,* advanced for nation-state actors, basic for hacktivists; (iii) the operational mode, *e.g.,* stealthy for nation-state actors, visible for hacktivists; and (iv) the access vectors, *e.g.,* legitimate internal access for insider attacks, supply chain attacks for nation-states. Those characteristics influence the quality and relevance of the formed threat hypothesis, their verification, and validation.

B. Threat Hunting Methodology

Considering the variety of threat actors and their motivations and intention, the threat hunting process starts by assuming the existence of adversaries in the system, and it aims at finding unusual behavior that may indicate the presence of malicious/suspicious activities [68]. The initiation of investigation typically falls into three main categories:

1) *Pro-active Investigation:* Pro-active investigations are usually triggered by the identification of new threats using a large pool of crowd-sourced attack data (*e.g.,* CVE – Common Vulnerabilities and Exposures, CVSS – Common Vulnerability Scoring System, MITRE ATT&CK) [69]. By using the system's telemetry data (*e.g.,* system logs, app logs, network traces), the hunter generates possible hypotheses for the attacker's activities using the attack knowledge base. The hunter then attempts to discover the existence of the attacker's presence and activities in the organization's environment to validate or refute those hypotheses.
2) *Investigation based on IoC/IoA:* This approach leverages tactical threat intelligence to catalog known Indicator of Compromise (IoC) and Indicator of Attack (IoA) associated with new threats [70], such as MITRE ATT&CK [71]. The triggering of an IoC/IoA drives the hunter to uncover potential hidden attacks or ongoing malicious activities in the organization's environment generally associated/related to the IoC/IoA.
3) *Investigation based on Analytics:* This approach merges powerful data analysis techniques and machine learning algorithms to analyze a myriad amount of data/information. Such a merger intends to efficiently detect abnormalities that may indicate the existence of potential malicious activities [72]. These abnormal activities become hunting leads that help security analysts to identify stealthy threats.

C. Threat Hunting Types

The hunting hypotheses/triggers are considered the starting point for an in-depth threat investigation. Based on the investigation nature, IBM [73] differentiate three types of hunting:

1) *Structured Hunting:* Structured hunting is principally established on the existence of an IoA and/or a TTP (Tactics, Techniques, and Procedures) [74]. Since all hunts are aligned and based on the TTP, the hunter can usually identify a threat actor even before the attacker can cause damage to the system. This type of hunt is built on a Cyber Threat Intelligence (CTI) framework such as Lockheed Martin Cyber Kill Chain [75] or MITRE ATT&CK framework [76].
2) *Unstructured Hunting:* Unstructured hunting is triggered based on a trigger of an IoC [77]. This trigger often drives the hunter to perform an investigation to uncover pre- and post-detection patterns. It also leads the hunter to examine as far back as the data retention and previously associated offenses allow.
3) *Situational Hunting:* Situational hunting, also known as entity-driven hunting, is initialized based on hypotheses generated from the internal reports (*e.g.,* risk assessment) or analysis of trends and vulnerabilities [78] and could be both structured and/or unstructured. The hunting process is predominantly led using crowd-sourced attack data and telemetries that help in revealing new TTPs of current threats that might stealthily exist in the organization system.

D. Hunting Maturity Model

In order to assess the effectiveness of a detection, hunting, and response system for an organization, security experts developed the hunting maturity model [18], [79], [80]. The higher the maturity level is, the more cybersecurity capabilities are ensured. The maturity model is divided into different levels:

- Level 0 - Initial (*Reactive*): The organization does not have established any procedures to detect and identify attacks and threats, and it mainly relies on ad-hoc responses.
- Level 1 - Minimal (*Proactive*): The organization has the fundamental procedures in place to detect and identify threats by using existing security tools for detection and investigation.
- Level 2 - Procedural (*Organized*): The organization has established formal methods and procedures to detect and identify attacks by integrating a dedicated team to take appropriate actions and responses.
- Level 3 - Innovative (*Analytical*): The organization has advanced threat hunting tools and specialized teams to analyze, identify, and respond to threats.

- Level 4 - Leading (*Proactive and Adaptive*): The organization has proactive, adaptive, and advanced threat hunting programs to foresee threats and adapt its defense actions accordingly.

The model has three factors to judge an organization's threat-hunting system, which are:

- *Data telemetry:* The quality and quantity of collected data used during/for threat hunting play a critical role in the overall hunting efficiency [81]. The higher the volume and the greater the variety of data provided to the system, the more insights the hunter will find and the more TTPs will uncover.
- *Visualization and analysis techniques:* Starting from the immense quantity of collected data, visualization techniques will help in better understanding entities, actors, and their activities in the organization. Analysis techniques, on the other hand, will facilitate the identification of new threat vectors and TTPs [82].
- *Automated analytics:* Data analytics requires a massive amount of time and knowledge expertise that might not be easy to purvey with the exponential increase of data [83]. Automating the analysis routine will not only help reduce the hunting time but also the required analyst qualification, in addition to enhancing analysts' insights, together with the scale and efficiency of hunts [84].

IV. Threat Hunting: General Architecture

In order to provide an efficient hunting performance, we design a top-down threat hunting architecture that represents and consolidates different threat hunting approaches in the literature [9]. As threat hunting is rather a new center of interest in academia, we believe this general architecture would help practitioners and researchers to better understand different approaches and help clarify gaps in the research. The designed architecture, depicted in Figure 2, has three main components: (i) data sources, (ii) threat hunting system, and (iii) security outputs. We detail these components in the following subsections.

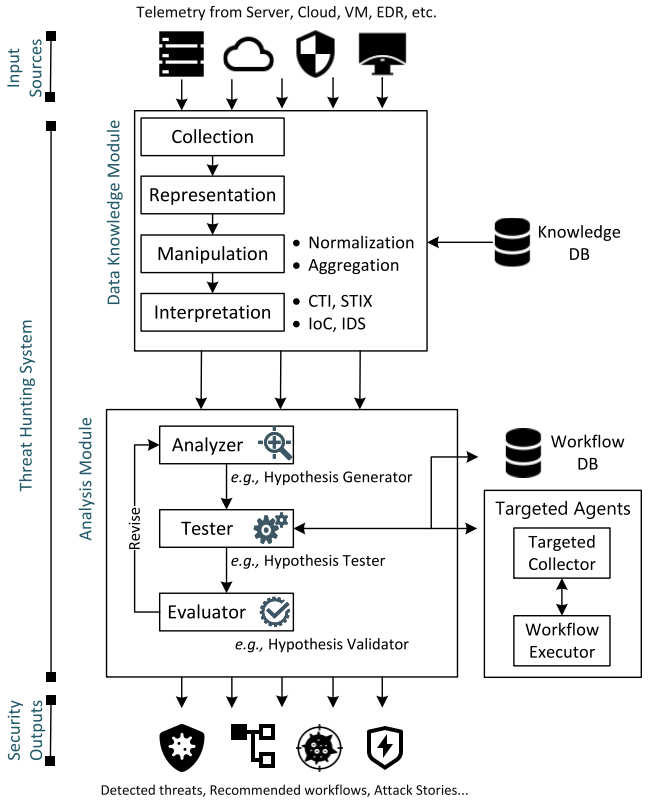

Fig. 2. Threat hunting general architecture.

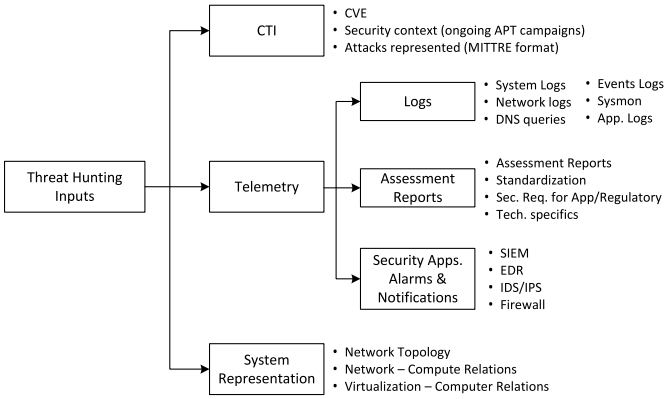

Fig. 3. Input data sources for threat hunting process.

A. Data Sources

Data is considered the primary element in the threat hunting process [85] and can be internally or externally collected from various sources [86], such as servers, cloud environments, virtual machines, Endpoint Detection and Response (EDRs), nodes, etc. Internal data refers to any sort of information, data, statistics, and trends that an organization owns or are generated by its users during the working routine. External data includes any kind of information, knowledge, and data generated outside the organization's boundaries. This data is usually validated by security experts and ready to use in the hunting process. The quality and quantity of collected data play a vital role in hunting efficiency and proficiency. Based on the origin of the data, we differentiate three paramount data inputs in threat hunting, as shown in Figure 3 and explained detailed below.

- *CTI:* It refers to any contextually enriched information on actors, threats, and vulnerabilities. The objective of CTI is to make better security-related decisions and improve security posture. In recent years, many companies start providing CTI streams to enterprises using different sources [87] (*e.g.,* IoC [88], CVE [89], CVSS [90]). Attacks and APT could be represented using MITRE ATT&CK [76] recommendation (*e.g.,* TTPs).
- *Telemetry:* Data telemetry might include: (i) *Logs:* including compute and network-related logs such as system logs, network, events logs, and Sysmon logs; (ii) *Assessment reports:* generated by internal or external security and analyst team, standardization security requirements, security requirements for applications and regulatory, and technical specifications; and (iii) *Security apps alarms and notifications:* any information generated

by security applications such as SIEM, EDR, IDS/IPS, and firewall.

- *System Representation:* represents the relations between different system components and the dynamic status of the system such as network topology, relations between network and computing entities, and virtualization computer relations.

B. Threat Hunting System

This component is the engine of hunting and has two main modules: (1) the data knowledge module that aims to collect and build required data for hunting and (2) the analysis module that uses the available data for investigating and discovering the threats.

1) Data Knowledge Module: The data knowledge module starts with the data collection process. This process begins by identifying what data needs to be collected and from what device/source [91]. Collecting data enables security teams to build multiple layers of knowledge when checking for threats and attacks as well as creating a secure sandbox environment. This environment does not help only in revealing potential vulnerabilities and risks within hardware and software-based systems such as networks, applications, routers, switches, and appliances; but also tests the infrastructure when going through red teaming or user emulation procedures [92]. Indeed, the extra level of intelligence-led security assessment allows security specialists to extensively test organizations' cyber resilience, threat detection, and incident response capabilities [93]. Since various tools could be used to collect data with different formats (*e.g.,* STIX Shifter, STIX Analyser, IBM Krestel, etc.), the data is then represented in (i) graphical formats: for monitoring and visualization purposes and to facilitate interpretation for the security analysts [94] and/or (ii) descriptive formats: such as in JavaScript Object Notation (JSON), eXtensible Markup Language (XML), and Trusted Automated eXchange of Indicator Information (TAXII) for storage and transmission purposes [95]. After that, the data is processed through different techniques known as data manipulation (*e.g.,* data normalization, aggregation, and transformation) [96]. Those techniques tend to alter, change, and translate data into the required format in order to be easily cleaned and mapped for extracting insights and facilitating the decision-making process. Finally, the data interpretation process takes place in order to review the analyzed data through some predefined processes, assign some meaning to the data, and draw relevant conclusions.

2) Analysis Module: Although the expertise and qualification of the hunter play a vital key in conducting successful threat hunting, it is essential to follow a formal hunting process to ensure consistency and efficiency [97], [98], [99], [100]. Cybersecurity experts follow a widely known hunting control loop (depicted in Figure 4) that consists of four main steps:

(1) *Hypotheses Creation:* based on the available data telemetry and observations, the hunter raises different questions that are considered investigation hypotheses [101]. The latter could be generated manually using the security analysts' expertise or automatically using risk management algorithms.

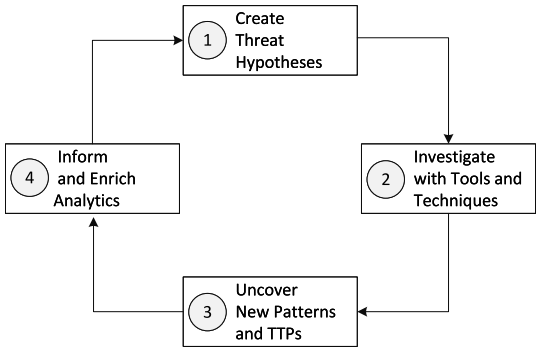

Fig. 4. Threat hunting control loop.

(2) *Investigation via Tools and Techniques:* Once hypotheses have been generated, the hunter starts the verification and validation steps using all relevant tools and techniques and the available data [102].

(3) *Uncover new patterns and TTPs:* The result of hypotheses verification/validation is to prove the existence of malicious/suspicious activities or refute it. When a hypothesis is proven (*i.e.,* the presence of malicious activity), the hunter needs to identify the threat infrastructure and related sequences, which also helps in uncovering new TTPs [88].

(4) *Inform and Enrich Analytics:* A successful hunt should be automated to maximize the efficiency of the hunting routine and minimize the investigation time [83]. This can be accomplished by (i) enriching the cybersecurity databases with the newly identified patterns and TTPs, (ii) developing a new analysis in existing tools, and (iii) providing feedback to the analyzer in order to enhance its accuracy. In fact, the faster the hunt can be automated, the fewer repetitions the hunters will have to go through, and the sooner they can devote their curiosity and skills to verifying new hypotheses.

The proposed architecture maps the four hunting control loop steps into three components:

(1) *Analyser:* is responsible for scrutinizing the data provided by the Data Knowledge Module. Based on the input data and the CTI knowledge, the objective is to generate a list of potential hypotheses activities and threats that try to escape the detection system. The generated hypotheses must be testable to be verified and validated.

(2) *Tester:* is responsible for investigating and analyzing each hypothesis generated by the analyzer. Data visualization, basic search and querying, data clustering, and ML techniques are mostly used to verify the hypotheses. The analyzer is enforced with a workflow database to generate workflow and targeted agents that employ softwarization to initiate targeted collectors in the virtual network/machines to gather additional artifacts and a workflow executor to assist hypothesis testing.

(3) *Evaluator:* is responsible for identifying malicious activities and attack patterns based on the results provided

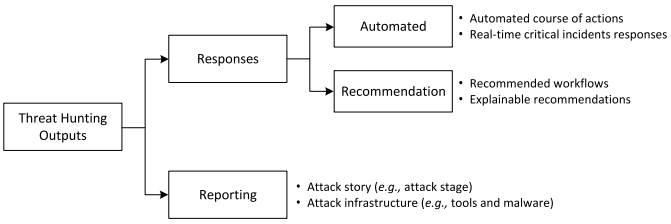

Fig. 5. Output outputs for threat hunting process.

by the Tester. Once a hypothesis is validated, the newly learned pattern/TTP is added to the hunting engine to help in detecting the attack in the future with fewer efforts.

C. Security Results

The proper implementation of security tools/techniques and exploitation of security expertise contribute to keeping organizations informed about the risks of advanced persistent threats, zero-day threats, and exploits, and how to protect against them. Figure 5 illustrates the security outputs produced by the merger of intelligent threat analysis and proactive threat hunting. To fulfill a specific security purpose and to facilitate automated security objectives, we logically group these capabilities into two main outputs: (i) *Responses:* it tends to provide automated responses (*e.g.,* course of action) in a real-time manner to different incidents as well as providing recommendations (*e.g.,* workflow) and explanations, and (ii) *Reporting:* contains pertinent information about the attack such as the attack story and attack infrastructure that could enrich the CTI knowledge base to detect/mitigate future attacks.

These outputs provide different security capabilities that contribute to the enforcement of the organization's security policy. These capabilities are groups of controls that support a common purpose and address common goals/objectives. We broadly assort those capabilities into three main categories based on NIST IR 8011 [131], as explained below:

- *Advanced Security Monitoring:* It also refers to security information monitoring (SIM) or security event monitoring (SEM). It aims to gather information and collect data on the organization's activities from different sources in order to process it using various analysis features. Since data originates from different sources and uses multifarious data representations and formats, advanced security monitoring applies a unified data representation to facilitate visualization, data storage, transferring, and streaming. Cybersecurity solutions use a wide range of representation schemes such as Ontology [132], graphs, JSON, XML, and STIX (Structured Threat Information eXpression [133].
- *Intelligent Threat Analysis:* The collected data is analyzed and filtered using various statistical methods and machine learning techniques in order to produce threat intels. The foremost objective of intelligent threat analysis is to discover, detect, and classify threats and anomalies in the system. It also assists in building the attack story and investigates malicious/suspicious activities using threat hunting.
- *Proactive Security:* Proactive security helps in enhancing and completing the existing intelligent security analysis and measures, as well as in achieving a high level of security awareness. It concentrates more on threat prediction, prevention, and mitigation rather than detection and response.

V. Threat Hunting Taxonomy

Based on our general architecture, we realize there are limited efforts around hypothesis analyzer and evaluator functionality. Most of the analyzers simply use an IoC to manually generate hypotheses. Similarly, the evaluation is conducted manually by security experts by investigating different logs and data while getting assistance from ML tools and techniques. The majority of threat hunting work revolves around the Tester, which is used as the main driver for our taxonomy. In this section, we review the existing threat hunting solutions. As shown in Figure 6, we classified the existing solutions using the used technique into four main categories: (i) ML/AI-based threat hunting, (ii) graph-based threat hunting, (iii) rule-based threat hunting, and (iv) statistical-based threat hunting. Solutions under each category have been further classified based on the approach. We also discuss the industry efforts under each category and provide some insights. Table II summarizes the reviewed solutions.

A. ML/AI-Based Threat Hunting

ML algorithms and AI techniques contribute to empowering, advancing, and transforming cybersecurity to the next level [134], [135], [136]. ML-based threat hunting [128] helps in enabling a more analytically advanced defense with minimum time and human efforts [137], [138], [139]. Data is a key player in informed-decision making and building strategic approaches that help to succeed ML techniques [140]. Supervised learning helps in training models to predict if events are malicious or not based on what the model has learned and classifying entities into malicious or suspicious nodes. Unsupervised learning assists in clustering data/events to similar groups based on different metrics (*e.g.,* behavior). To date, many efforts have been proposed to integrate ML/AI to enhance the threat hunting process [141]. Based on the technique used, we further group these solutions into three main classes: supervised learning, unsupervised learning, and reasoning techniques.

1) Supervised Learning: Various efforts use supervised learning techniques to discover threats based on different observations the model has trained. For instance, Alsaheel et al. [124] designed a framework that aims at constructing an end-to-end attack story starting from audit logs. The authors' main assumption is that different attacks may share similar abstract patterns/strategies regardless of the vulnerabilities exploited and payloads executed. The designed framework incorporates causality analysis, natural language processing, and machine learning techniques to produce a sequence-based model. Using Long short-term

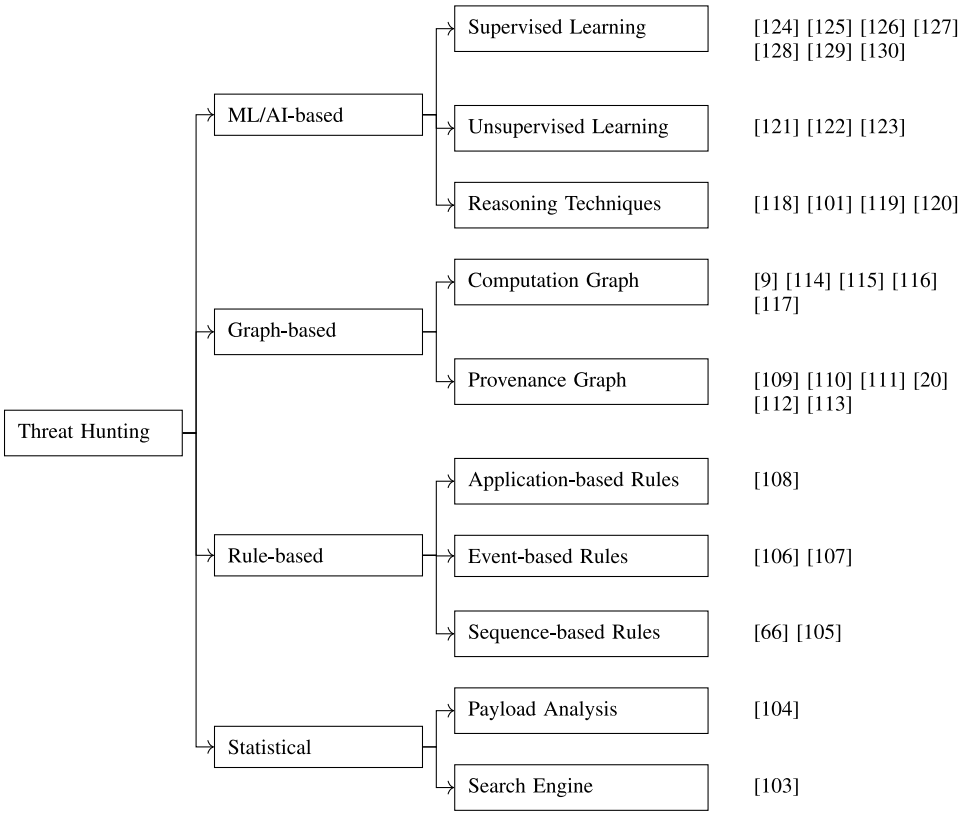

Fig. 6. Taxonomy of threat hunting solutions based on the used technique and approach.

memory (LSTM) and Convolutional Neural Network (CNN), the authors build a sequence-based model that seeks to identify patterns of attack and non-attack behaviors from a causal graph and to capture the stealthy nature of attacks. The model uses a Dropout layer for regularization, a Conv1D layer for sequence lemmatization, and a fully-connected layer with sigmoid activation to predict the attack probability of the sequences. After generating the threat alert events manually, the framework uses a causal graph to construct a sequence process graph and identify symptom nodes. Next, it groups a set of candidate sequences associated with the symptom node in order to identify nodes in a sequence that contribute to the attack by using the sequence-based model. In doing so, the framework is able to construct an end-to-end attack story using sequence events, however, the model may fail to hunt/detect new attacks with new patterns/strategies that did not appear in the training dataset. Similarly, Afzaliseresht et al. [125] designed an automated log-driven storytelling report generation mechanism. The mechanism operates in multiple layers, including (i) preprocessing layer: to collect log files and extract basic fields (*e.g.,* date, time, IP address), (ii) extraction layer: to build expressive knowledge from log files using binary search and correlation between time, source/destination IP address, and alert message based on Snort rules, (ii) inference layer: to process log files by applying rule-based inference to extract malware evidence and definition. The evidence knowledge helps to generate the attack story by crawling security websites to search for the malware definition (*i.e.,* malware classification phrase) and k-means algorithm for clustering and malware classification, and (iv) story layer: to generate an attack story from the analyzed data. The authors applied artifact metadata and basic machine learning techniques to reconstruct past events and identify actors, riskiness, and evidence of events in relevant logs. Local and global information is used as input to the model along with security experts' knowledge and expertise. Villarreal-Vasquez et al. [126] focused on hunting insider threats using time series log events. Using system event sequences collected from several endpoints, the authors designed an anomaly detection framework based on LSTM. The model has four modules for data collection, data selection, model generation, and anomaly detection. After collecting data, the proposed solution changes the event feature vector generated by the EDR agent to meet a set of selected features. Features, on the other hand, are a design choice that controls the granularity and the total number of possible events. Weak features might impact the overall performance if not well-defined and bounded. The detector uses a vocabulary of events for feature selection (*e.g.,* transformation function) and generates the model using either by-machine or by-time splitting schemes (*e.g.,* 80% for training and 20% for validation). The model creates behavioral profiles of various applications by computing the probabilities of each event and then learns the expected event patterns in a computer system in order to recognize attack sequences.

Lin et al. [127] designed an attack tactic labeling scheme in order to determine the current attack state and infer its purpose. The authors combined similarity-based labeling and ML-based labeling on top of the MITRE ATT&CK framework. The

TABLE II
SUMMARY OF THREAT HUNTING SOLUTIONS BASED ON THE USED TECHNIQUE AND APPROACH

Domain	Ref.	Approach	Input Data	Data Representation	Security Capabilities	Security Outputs
ML/AI-based Threat Hunting						
Supervised Learning	[124]	Sequence-based model	Audit logs	Causal graph	Attack story	/
	[125]	Narrative analysis	Local and global logs	Raw logs	Attack story	Reporting
	[126]	Sequence analysis	Log files	NA	Threat detection	/
	[127]	Similarity-based labeling	IDS logs	Raw logs	TTP labeling	/
	[128]	Attack score calculation	Log files	Raw logs	Threat detection	/
	[129]	Hunting using SDN and ML	Log files	NA	Threat detection	/
	[130]	Federated learning	I/IoT traffic	NA	Threat detection	/
Unsupervised Learning	[121], [122]	Natural language processing	Security report	Graph	Threat identification	/
	[123]	Natural language processing	Threat description	Graph	Query generation	/
Reasoning Techniques	[118]	Automated investigation	CTI logs	/	Threat hunting	Workflow generation
	[101]	Evidential reasoning	Log files	Graph	Hypotheses validation	
	[119]	Reinforcement learning	Events sequence	Time-series	Attack detection	
	[120]	Logic-based deductive inference	CVE/CVSS, network topology	Ontology	Risk analysis Threat actor profiling	/
Graph-based Threat Hunting						
Computation Graph	[9]	Domain-specific graph computation	Log files	Graph	Hypotheses validation	/
	[114]	Community detection	Apps logs	Graph	Attack story	/
	[115]	Graph analysis	IoC database	Persistent graph	Reduce false-positive alerts	/
	[116]	Technique attack template	CTI reports	Attack graph	Identify attack techniques	/
	[117]	Correlation analysis	Apps logs	Graph	Generate attack routes	/
Provenance Graph	[109]	Natural language processing	CTI reports	Provenance graph	Attach story	Reporting
	[110]	Causal analysis	Security alerts	Provenance graph	Threat detection Attack steps	
	[111]	Graph analysis	Alert/IoC	Knowledge Graph	Reactive hunting and detection Vulnerability analysis	/
	[20]	Link prediction	IoC	Knowledge Graph	Similarity-based hypotheses generation	/
	[112]	Causal analysis	Control/Data plane activities	Provenance graph	Root causes	/
	[113]	Multi-level provenance analysis approach	Log files	Provenance graph	Root causes	/
Rule-based Threat Hunting						
Application-based Rules	[108]	Hunting with GRR Rapid Response	Logs	Raw logs	Threat hunting	/
Event-based Rules	[106]	Rule-based detection and scoring	Dropbox login activities	Login graph topology	Lateral movement detection	/
	[107]	Statistical analysis	Network traffic/statistics	Raw logs	Threat analysis	/
Sequence-based Rules	[66]	Evidence-based knowledge	Sysmon logs	Ontology	Threat level classification	/
	[105]	Temporal word embedding	IPS telemetry	Sequence	Vulnerability context understanding	/
Statistical-based Threat Hunting						
Payload Analysis	[104]	Adversary emulation	Security reports	NA	Threat detection	/
Search Engine	[103]	Statistical analysis	Network traffic/statistics	Graph	Threat analysis	/

scheme consists of two main phases: (i) similarity-based tactic labeling, which tends to assist experts with tactic labeling and accumulating the dataset quickly, and (ii) ML-based tactic labeling to learn features of different tactics from the expanded dataset. For the former, the authors used Suricata rule-based engine that employs Keywords for Tactic Matching function (*e.g.,* the message field) to find a match between the attack and tactile, while the latter applies Term Frequency – Inverse Document Frequency (TFIDF) vectorization to calculate the importance of terms in each tactic through term frequency-inverse document frequency algorithm and then calculate the similarity with each tactic. Chen et al. [128] designed a semi-automated threat hunting system using existing machine learning algorithms. The designed model tends to detect attacks and helps in hunting APTs. A supervised classifier has been designed using support vector machines, decision trees, and a neural network. Starting from an event alert, the classifier recommends if the event is malicious or benign. The model has been enhanced to detect anomalies by associating each anomaly tag to a cluster using Local Outlier Factor (LOF), Isolation Forest (iForest), and Density-Based Spatial Clustering and Applications with Noise (DBScan) algorithms. The final version of the model uses graph representations of events to create a more precise annotation. The model uses Community Detection, Attack Score Calculation, and Label Propagation to identify malicious events. Schmitt et al. [129] presented a high-level intelligent threat hunting system using the advantages of software-defined networking (SDN) in

conjunction with ML techniques. Taking the benefits of the programmability in SDN to train ML models using normal network behavior, the authors reviewed different models, such as (i) the Frequent Pattern Mining model to train unlabeled network data using a logging system, (ii) the Gradient-Boosted Tree model to predict the outcome of a new incoming event using historical network data, and (iii) the SDN Response System to build a queue of action (*e.g.,* routing control, traffic dropping, or blacklisting). New rules can be created or updated based on the model outputs, which help mitigate threats on the network while upholding maximum network uptime. As long as data privacy is concerned, Abdel-Basset et al. [130] designed a federated deep learning model for hunting cyber threats. The model captures the temporal and spatial representations of network data (*e.g.,* events) and applies the federated learning concept to train the global model without sharing the device's data. Each edge server trains a classification model locally using an adaptive stochastic gradient descent algorithm, while the cloud server aggregates the model to provide better accuracy with fewer data exchanges and preserving privacy. The classification model uses AutoEncoder (AE) and Gated Recurrent Unit (GRU) architecture inspired by a Recurrent Neural Network-based AE. The authors also introduced a micro-service placement algorithm to choose at what device the threat hunting model should be placed in order to minimize the computational resources of the participants.

2) Unsupervised Learning: Gao et al. [121], [122] designed an evidence-based knowledge system that helps in extracting knowledge about threat behaviors (IoCs and their relationships) from unstructured Open-Source Cyber Threat Intelligence (OSCTI) reports in order to facilitate the threat hunting process. Toward this, the authors introduced an unsupervised natural language processing (NLP)-based system that extracts IoC entities and relations and then builds a threat behavior graph. The solution applies (i) block segmentation to extract IoCs and their relations from each block to build threat behavior graph and (ii) IoC recognition and protection techniques using regex rules extracted from CTI report to identify different types of IoCs.[1] The authors also introduced a domain-specific query language and query synthesis to interrogate the log data stored in the database. Both PostgreSQL and Neo4j databases have been used for relational storage and graph storage, respectively. The query language is a declarative language based on event pattern syntax, that integrates a collection of critical primitives such as system entities and relations as pillar elements to provide explicit construction for entity/event types, operations, and path patterns. Karuna et al. [123] designed an architecture that aims to automate query generation for threat hunting. The designed architecture has multiple modules and uses domain-specific language (DSL) for the threat description database MITRE ATT&CK framework to map TTPs. By using genetic programming to perform genetic perturbations of IOCs, the TTP Store module defines all possible combinations of the abstract workflow in DSL, while the Query Scheduler module generates queries and stores them in the Data proxy. The latter ingests the confirmed (or refuted) hypotheses as a graph with TTP-based inferences. The Threat Matcher module uses natural language processing (NLP) to extract threat descriptions using pattern recognition and signature-based methods and then match them with the defined data-store to generate outputs in different formats (*e.g.,* Markdown, JavaScript).

3) Reasoning Techniques: Some attempts have been presented to automate the hunting process, draw conclusions, and construct explanations using existing knowledge. For instance, Puzis et al. [118] introduced a high-level framework for semi-automated threat hunting and forensic investigation. The authors combined different CTI into one centralized database and proposed two main modules: (i) attack hypotheses generation module that tends to, starting from existing knowledge, generate and rank attack hypotheses during the investigated incident relying on distance metrics with known IoCs, and (ii) workflow generation module that produces ready-to-execute distributed workflows. The designed architecture is promising in providing proactive and automated threat hunting using workflows and re-programmable security agents for execution. However, the authors did not discuss its applicability in real-world scenarios nor evaluate its evaluation. Araujo et al. [101] conceptualized the implementation of an evidential multi-criteria cyber reasoning framework for threat hunting. The authors introduced three-dimensional sub-spaces: (i) knowledge: to represent what is known about the internal and external environments, (ii) hypothesis: to define domains of detection hypotheses and threat hypotheses, and (iii) action: to represent the defense mechanisms and protective measures. The subspaces are represented in a case graph format. The cyber threat hunting model conceptualizes threat detection, analysis, and response as first-class components, where the authors used the existing knowledge to verify and validate the initial hypothesis. To achieve this, they conceived the Threat Hunting Description Language and Graph Query to facilitate the creation and execution of workflows (*i.e.,* hunt programs). Dehghan et al. [119] introduced a deep reinforcement learning system to assist the security team by projecting the next step of an APT. Modeling the attack kill chain as a time series and relying on the fact that all attack steps in the kill chain have known relations, the authors employed LSTM to approximate the best action of each step and then predict the attacker's next move. The proposed solution converges using reinforcement learning features (trial, error, and learning) by selecting the optimal next move based on the previously seen attack kill chains. Yet, the proposed solution will not be able to predict the next move for a never seen attack (*i.e.,* new APT, new attack pattern, zero-day attack) in the previous time-series. Qamar et al. [120] employed Web ontology language (OWL) to provide a formal specification, semantic reasoning, and contextual analysis for threat analytics. The ontology aggregates different inputs (*e.g.,* STIX, network, CVE, and network topology) to provide a unified knowledge space that operates using Protégé.[2] The reasoner engine has been designed using

[1]IoC Parser: https://github.com/armbues/ioc_parser

[2]Protégé: https://protege.stanford.edu/

logic-based deductive inference rules. Rules follow the horn-clause style, defined in semantic Web rule language (SWRL) language, and applied on top of the defined ontology by performing mapping and conflict detection. The objective of the mapping is to improve the capability of timely threat identification, provide semantic reasoning, and produce risk analysis and threat actor profiling.

4) Summary and Insights: Today's security tools are competitively racing to implement ML/AI and data-driven schemes. The diversity, availability, and meaningful information of data contribute to enhancing prediction and prevention capabilities. Various security tools, *e.g.,* Cortex XDR [142], QRadar [143], and Splunk [144] implement different ML techniques to enrich incident engine, response capabilities, and threat intelligence feed. ML algorithms intensely rely on the data for training. A large amount of data does not always train an efficacious model. In fact, diverse and quality data can lead to better production of a more precise model. Also, due to the benign diversity of activities of applications, system administrators and developers, in addition to users across the organization, the definition of normal behavior is often a challenging exercise and might lead to generating false-positive alerts in the threat hunting process. However, given the sheer size of collected data and the large networks considered, an automated approach to detect and hunt down the attacks in such networks become primordial for modern security approaches, *i.e.,* there is a need to filter, select, and provide the best information for human security operators. ML/AI, therefore, can play an essential role to automate threat hunting. Supervised learning helps in predicting events with high accuracy but might fail to recognize rare events. Unsupervised learning leads to inaccurate results without experts' intervention. ML techniques often do not allow traceability of decisions/conclusions. In recent years, there have been many efforts for explainable ML/AI to address this shortcoming [145], [146]. However, there are still efforts necessary to cover all ML/AI used methods. Machine reasoning, on the other hand, can provide the possibility to trace the cause of conclusions, but it requires the definition of huge data knowledge, rules, and fine logic that need expensive inputs from experts.

B. Graph-Based Threat Hunting

Graph-based threat hunting merges graph representation and theory to provide threat analysis and investigation [42]. Graphs are usually directed to respect events' order and semantics, where system-level entities are presented as verticals and relations as edges. The graph-based approaches such as graph-feature-based similarity approach, node behavior extracting, and decomposition-based event [147] are used to analyze threats, detect anomalies and root causes, identify involved nodes/data, and understate patterns in data breaches [148].

1) Computation Graph: A computation graph is a directed graph used to describe computational logic. The nodes correspond to operations or variables, and edges represent function argument (*i.e.,* dependency), where a node with an incoming edge is a function of that edge's tail node [149]. Shu et al. [9] introduced threat intelligence computing, where they modeled threat discovery as a graph computation problem. The objective is to (i) eliminate heterogeneous data representation and (ii) enable efficient programming to solve the problem. The computation graph has element attributes (*e.g.,* event type), element relation (*e.g.,* read event), and security knowledge (*e.g.,* anomaly score). To facilitate threat hypotheses validation, the authors introduced a domain-specific graph computation language, τ-calculus, evidence mining, and data inspection. Hypotheses are manually generated by security experts and then translated to τ-calculus programs using domain-specific language and then evaluated using computation graphs. Pei et al. [114] designed a log-based intrusion analysis system to understand and discover the attack trace. The authors modeled the problem as a community discovery problem and adopted graph analytics (*e.g.,* social network analysis) with ML algorithms (*e.g.,* feature selection) to build correlations between log entries where they conservatively treated any log entry as malicious if there exists a path from a tainted attack-related to the log in question. Causality analysis has been leveraged to build a relationship between multiple logs in the system. Social network analysis and community detection algorithms have been used in the weighted graph settings to automatically generate a uniform vector representation and discover the attack trace. The analyzer repeats the Louvain method to greedily optimize the local modularity as the algorithm progresses and calculate the degree of the density of the connections within communities. Berady et al. [115] formalized the threat hunting process conducted by an incident response team in order to evaluate the efficiency of a detection chain. The proposed model tends to analyze both the attacker's propagation and the defender's knowledge that are modeled through a graph of objects and components. Towards this, the authors employed the STIX standard to define observable objects and information, which are presented in a persistent graph. Using graph techniques, such as satisfaction relations and small/big step propagation as well as highlighting events of interest from an IoC database and a set of detection rules, the authors were able to detect events of interest that help, by consequence, in minimizing false positives. Focusing on threat intelligence, Li et al. [116] used the diversity of CTI reports to extract structured threat intelligence and then identify attack techniques. The authors used a learning-based named entity recognition (NER) model to extract attack entities and dependency trees for each sentence with a learning-based nature language parsing model to extract attack dependencies. The attack behavior is then modeled as a graph, with nodes representing attack-relevant entities and edges representing dependencies. The authors also introduced a technique knowledge graph (TKG) concept that summarizes causal techniques from attack graphs. The designed solution applies a graph alignment algorithm, with knowledge expertise (*e.g.,* MITRE ATT&CK knowledge base), to calculate the alignment score between the technique template and attack graphs. The latter helps in identifying a technique template in order to determine and aggregate the technical level of threats across reports and then describe the complete attack chain. Jadidi and Lu [117] devised a three-phase threat hunting

framework tending to detect threats against industrial control system devices in the earliest phases. The first stage consists of identifying events or any external source that can trigger the hunting stage, which could be an IoC, IoA, or any external information provided by security analysts. The authors used a two-dimensional graph that presents the cause-and-effect relationship between events. After the trigger, the hunting process starts, which combines MITRE ATT&CK Matrix and Diamond model for analysis and hypothesis validation. The authors used correlation algorithms to discover correlations between events and logs and the HELK tool[3] to identify adversarial TTPs and visualize the attack routes. The last stage consists of generating new IoC that could be used for future threat hunting.

2) Provenance Graph: A provenance graph is a directed acyclic graph where each node has a set of features describing the process or data it represents (*e.g.,* timestamp, description). The goal of provenance is to assist users in understanding their data [150]. Satvat et al. [109] designed a framework that manages to extract the full picture of the attack behavior (threat information extraction) from the technical reports in the form of a provenance graph. To facilitate the process, the authors used various transformation techniques on text to build a simple representation. Indeed, the authors applied a set of natural language processing techniques to transform a complex text into a more simple and consumable representation, such as (i) Tokenization, Homogenization, and Conversion to transform long and complex sentences into shorter sentences appearing in a canonical form; (ii) Ellipsis Subject, Pronoun, and Entity Resolution to reconcile implicit references that refer to the same entity into the actual referent; (iii) Sentence and Word Verbosity to downsize verbosity and obtain a concise description of the attack behavior; and (iv) Semantic Role Labeling to extract the attack behavior including subject, object, and actions. Bhattarai and Huang [110] adopted a provenance graph-based threat hunting system to detect and reconstruct APT campaigns across multiple hosts. In order to minimize the required human effort to correlate security events, the proposed solution takes the alerts generated by sophisticated intelligence and anomaly-based systems (*e.g.,* EDR) and applies causal analysis to extract the group of events that are most likely to represent the attackers' activities (based on the temporal and causal relationship). The analysis is performed per host and then a hierarchical graph traversal is performed to conduct cross-host attacker activity correlation using the prize-collecting Steiner Tree [151]. This correlation helps in detecting the compromised entities, reconstructing the attackers' steps, and abstracting them into an attack graph. Pelofske et al. [111] proposed the use of graph databases to perform threat hunting and vulnerability analysis on top of open-source intelligence. The open-source intelligence is built using different structured and unstructured sources (*e.g.,* blogs, social media, threat intelligence fees, MITRE ATT&CK, CVE, etc.) where the security expert can perform different graph-based algorithms (*e.g.,* page rank) to drive the hunter. Although graph-based algorithms databases allow efficient and comprehensive analysis of relationships between different entities and events in the graph that helps to identify more complex attack patterns, the process is still time/resource consuming since the hunting is conducted in a reactive manner by relying on the security expert to analyze and deduct different IoC and patterns. Kaiser et al. [20] focused on inferring ATT&CK techniques from a set of observable artifacts. Toward this, the authors combined data from multiple threat intelligence sources to associate high-level ATT&CK techniques with low-level telemetry. They also presented a mechanism to generate an attack hypothesis relying on knowledge graph traversal algorithms (*e.g.,* count paths) and link prediction methods (*e.g.,* Term frequency-inverse document frequency). Based on the occurrence of an IoC, the proposed solution calculates the score between the existing observable and the received techniques. Initial hypotheses are generated based on the graph traversing algorithm and then refine the hypotheses by predicting the link between observable and known techniques. Although the application of this solution is promising, the effectiveness and robustness of the proposed algorithms are questionable. Focusing on attack root causes, Ujcich et al. [112] designed a provenance-informed causal observation tool tending to solve the dependency explosion problem for SDN attack provenance. In order to identify the root causes of attacks and understand their scope/effects, the authors leveraged a fine-grained provenance model. The model leverages (i) app event listeners to partition data and process execution and (ii) a data plane's topology to indirect control plane activities caused by data plane packets. The model helps in constructing past control and data plane activities and then identifies root causes using the iterative backward-forward trace and common ancestry tracing algorithms. Similarly, Tabiban et al. [113] addressed the root cause identification of security incidents in a network functions virtualization (NFV) environment. The authors presented a multi-level provenance solution that captures the dependencies between NFV levels with higher-level semantics. The designed solution links the provenance graphs at different levels of the NFV stack by capturing the cross-level dependencies between different abstractions of the same network service. By doing so, the authors were able to perform cross-level mapping and hence trace a security incident back to its root causes located at a different level. The authors adopted different techniques such as multi-level pruning, mining-based aggregation, and rule-based natural language translation to further improve the solution's interpretability. These solutions build graphs with different entities and relations to represent insightful information and events, and then execute graph search algorithms to find the root causes of attacks based on available knowledge (*e.g.,* IoC databases).

3) Summary and Insights: Security tools that implement graph-based analysis have a solid mathematical basis and search, which would maximize the ability to discover potential threats and mission impacts while minimizing the time needed for organizing multiple disparate data sources into meaningful relationships. The objective of these tools (*e.g.,* Cisco XDR [152], QRadar [143], Splunk [144]) is to bring isolated data and events together in one piece for efficient analysis

[3]The HELK: https://github.com/Cyb3rWard0g/HELK

and decision-making to improve the overall network security posture. Graph-based threat hunting maps the potential attack paths through a network that aims to improve the organization's security. Although a graph-based solution focuses on the protection of mission-critical assets and maintains situational awareness, it requires optimized graph algorithms to bind the immense expansion of graph nodes/edges and to accelerate knowledge extraction and analysis. Although aggregation and pruning algorithms can help in reducing expansion dilation, they suffer from deviating information or breaking the semantics.

C. Rule-Based Threat Hunting

Rule-based threat hunting [153] describes a set of security rules (*i.e.,* policies) that identify malicious nodes or activities using various attribute-value knowledge representations. Rules can be written based on applications (*e.g.,* behavior, traffic), events, and sequences to identify any malicious activity. The hunting process relies on comparing the sensitivity of the objects being accessed to the corresponding attributes aiming to identify suspicious/malicious activity [154].

1) Application-Based Rules: Application-based rules [155] tend to specify the action to be performed when the traffic/behavior/signature of an application matches the associated match criteria. Rasheed et al. [108] discussed the deployment of the threat hunting process using GRR Rapid Response.[4] The latter is an open-source incident response framework that performs scalable remote live forensics for host applications running different operating systems using the hunt feature and matching rules. The authors tested the framework using two use cases: memory corruption exploits with a reverse TCP (Transmission Control Protocol) shell payload and persistence mechanism achieved by a client-side exploit.

2) Event-Based Rules: Event-based rules [156] are used to discover attacks and threats based on the existence of predefined events, mostly from log files. These rules follow the occurrences of the event to hunt down the attacks. Ho et al. [106] designed a system for hunting and detecting lateral movement based on commonly available enterprise logs. The idea is based on constructing a graph topology of login activities and identifying suspicious sequences of logins that correspond to lateral movement. In doing so, the authors utilized an inference login causality algorithm along with a set of detection rules and a specific-based anomaly scoring algorithm to determine the broader path(s), minimize false positives, and then identify lateral movement. Radoglou-Grammatikis et al. [107] concentrated their efforts on the deployment of honeypots in an industrial IoT environment in order to enhance threat hunting capabilities. The idea consists of providing an analyzer server capable to (i) collect data from multiple honeypots applications and performs traffic aggregator using Logstash, (ii) store data in a Security Events Database using Elasticsearch technology, and (iii) provide a visualization engine for hunting purposes using Kibana query language. The authors employed Thompson Sampling, a reinforcement learning method, to find the optimal number of honeypots to be deployed in order to provide maximum protection, taking into account the available computing resources and the behavior of the attacker. The authors then applied statistical analysis of the honeypots' data collected through various log files.

3) Sequence-Based Rules: A hunting policy is defined as a set of rules for retrieving a sequence of IoC/IoA fingerprinting a given attack/threat. Mavroeidis and Jøsang [66] designed an automated threat assessment system by providing an analysis of continuous incoming feeds of Sysmon logs. The log information is stored in cyber threat intelligence ontology (CTIO) [157], which is used for threat assessment and classification. The authors then used evidence-based knowledge to provide situational awareness of potential IoC. They also utilized OASIS OpenC2 to automate the course of actions in the case of threat detection, however, both detection and automated response are not clearly detailed in the solution. Shen and Stringhini [105] tended to develop a proactive defense line by understanding the context in which a vulnerability is exploited and detecting its unforeseen context changes. Starting from collected telemetry, the authors employed temporal word embedding on each security event and sequence to convert it into a word and sentence, respectively. Pointwise mutual information is then applied to calculate the co-occurrence of any two events within a context window aiming at understanding the modus operandi of attackers.

4) Summary and Insights: Rule-based threat hunting is implemented in most commercial security tools (*e.g.,* Helix [158], Cortex XDR [142], Splunk [144]). Despite the common objective, some SIEM/(X)SOAR solutions come with customized features, such as: (i) application-specific rules: control the application's connection and traffic by restricting its traffic and payloads; (ii) event-based rules: bound the CTI's behavior as a result of an incoming event; and (iii) context-based rules: rely on the network environment and context of the flow/application. Each rule can describe a specific context and be associated with a behavior. Rule-based hunting is mostly suitable for situations where consistency is critical. However, they do not scale well when it comes to situations where there are automatic adjustments to changing conditions. These include rapid expansion or new application programming interface (API) connections with third parties. This is due to the static nature of rules from the deployment and working perspective. The quality of rules also depends heavily on the expert's quality, expressiveness, and completeness of the rule language compared to the expertise domain and also on the quality of the capture of the experts' knowledge. Furthermore, rule-based schemes may cause unacceptable productivity impacts for end-users (*e.g.,* automatically locking out users who fail to match some rules).

D. Statistical-Based Threat Hunting

Statistical-based threat hunting tends to combine statistical analysis and cybersecurity tools to identify suspicious entities and malicious activities, and therefore protect against cyberattacks and activities [159]. A conceptual understanding of the

[4]GRR Rapid Response: https://github.com/google/grr

vulnerabilities from a statistical perspective helps to develop the set of modern statistical models (*e.g.,* single/group profile generation, events clustering, entropy- and principal component analysis) and bridge the gap between cybersecurity and abstract statistical knowledge [160].

1) Payload Analysis: The MITRE ATT&CK framework could also be combined with statistical models to discover and detect documented adversary behavior [161]. For instance, Ajmal et al. [104] designed a threat hunting system using an adversary emulation model combined with the MITRE ATT&CK Matrix. The authors built hybrid strategies for attack emulation and used different offensive security payloads (*e.g.,* embedded VBA scripts in doc, document with hidden payload at EOF, malicious DLLs, PowerShell malicious script). For each threat, the authors considered the severity, progression, and relevance. The designed solution has multiple phases to create and validate hypotheses, capture and understand attack patterns by analyzing the payloads using the identification of data sources and analytical TTP's from the ATT&CK Matrix using similarity measure, and then uncover new TTPs in the system.

2) Search Engines: Search engines can be used to query different IoC, IoA, or events to discover the existence of threats in the system [162]. Elasticsearch technology (*i.e.,* Elasticsearch, Logstash and Kibana),[5] in particular, allows a huge volume data storage, searching, and analyzing in a near real-time [163], [164]. Almohannadi et al. [103] used the Elasticsearch technology to automate the hunting process and analyze honeypot log data and then identify the behavior of attackers, which ends up by finding attack patterns. Elasticsearch technology uses a bunch of analyzing tools/models based on users' needs, such as statistical rarity, audit logging, and field- and document-level security.

3) Summary and Insights: Statistical-based methods are widely used to identify threats and outliers with respect to a pre-defined model. Security solutions, *e.g.,* ArcSight ESM [165], Falcon XDR [166], Cisco XDR [152], come with different statistical models, such as Z-Score, Interquartile Range, and Boxplot, that could be used to analyze data and discover/detect threats based on the complexity of data and sophistication needed. These models aim at correlating and analyzing threat data as well as prioritizing, hunting, and remediating risks. Statistical-based threat hunting approach conventionally suffers from high false-positive rates, which might lead to security fatigue. It also requires significant investment in large-scale data collection, representation, and processing. It does not always provide enough contextual information, which can make analytic refinement challenging.

VI. Threat Hunting & Standardization

Sharing cyber threat intelligence is a critical component of automated threat hunting, which tends to enhance the protection level of organizations from various threats [167]. The objective of threat hunting is to uncover new threats and attacks in the enterprise through importing CTI and related information from internal security appliances and external data sources. In order to enhance the hunting capabilities, the threat hunting framework needs to be able to share the uncovered TTPs with other organizations and security solutions. Therefore, there is a need to support standard ways to structure, format, and represent data and knowledge. Although ISO (International Organization for Standardization) did not provide any standards/specifications related to threat hunting [168], it developed a number of standards relevant to threat hunting [169], [170], [171], [172]. Table III summarizes some of those standards. In the following, we discuss the standardization efforts for data formatting, including all of the representation, query languages, operational languages, and CTI sharing platforms.

TABLE III
Summary of the Current Status of Standardization

ISO	Description	Security Capabilities			
		Security Risk	Incident Mgmt.	Security Mgmt.	Risk Mgmt.
ISO/IEC 27001	Framework on how to establish, implement, maintain, and continually improve information security management	✓	✓		
ISO/IEC 27002	Code of practice for information security management such as initiating, implementing, and maintaining information security management		✓		
ISO/IEC 27035	Guidelines for information security incidents including detection and response to security incidents			✓	
ISO/IEC 27005	Guidelines for information security risk management including risk assessment and treatment				✓

A. Representation

Data representation is an essential component of the cybersecurity routine. Although there are many efforts to provide efficient data representation, fewer efforts have been presented from the standardization side [17]. For instance, Structured Threat Information eXpression (STIX) [133] is a standardized XML programming language used mainly to share and structure cyber threat information. STIX is maintained by OASIS and was initially sponsored by the United States Department of Homeland Security (DHS). The paramount target of STIX is to convey data about cybersecurity threats in a common and standardized language that is comprehensible by humans and security technologies. Indeed, threat analysts can integrate STIX in different use cases, including, but not limited to, (i) reviewing cyber-threats and threat-related activities, (ii) identifying patterns that could indicate cyber threats, (iii) facilitating cyber-threat response activities (*e.g.,* prevention, detection, and response), and (iv) sharing of cyber threat information within an organization and with external partners.

B. Query Languages

Providing a common language for hunting is an important step toward facilitating the hunting process. An efficient common language enables interoperability across a range of cybersecurity tools and applications and provides

[5]Elastic https://www.elastic.co/

a united understanding among security analysts. In addition, a common hunting language will help in automating machine-to-machine communication and processing. Various efforts have been presented by the open-source community to design an efficient yet standardized cybersecurity language. For instance, multiple companies and organizations collaboratively launched the Open Cybersecurity Schema Framework (OCSF) project [173]. OSCF[6] aims at helping organizations detect, investigate, and prevent cyber-attacks faster and more effectively by normalizing security telemetry across a wide range of security products, services, and tools. In doing so, OCSF schemes and procedures can be used in any environment, application, or solution. OCSF taxonomy maintains 6 fundamental constructs: data types, attributes and arrays, attribute dictionary, event class, category, profile, and extension. Each constructor has a set of primitives that facilitate the scheme definition. Although OCSF's taxonomy and MITRE ATT&CK [76] are two different projects, they do complement each other. Indeed, Categories are similar to Tactics, Event Classes are similar to Techniques, and Profiles are similar to Matrices. Yet, event classes are in only one category, while MITRE ATT&CK techniques can be part of multiple tactics and MITRE ATT&CK procedures can be used in multiple techniques [174]. MITRE ATT&CK has sub-techniques while OCSF does not have sub-event classes.

OASIS designed OpenC2 [175], which is an open-source standardized language that helps in supporting cyber defenses via a command-and-control technology.[7] Adopting OpenC2 language allows enterprise-wide interoperability of cybersecurity policy orchestration and therefore improves incident response to threats. An OpenC2 producer acts as a software orchestrator for many commands and is able to talk to multiple different consumers (*e.g.,* security units). Each OpenC2 consumer likely has multiple actuators and uses, at least, one of them to perform the action on the target. IBM Research introduced Kestrel [176], a threat hunting language, seeking to simplify and accelerate the threat hunting process. Kestrel uses STIX and runs on top of STIX-Shifter[8] to automatically compile threat-hunting steps down to other programming languages that different data sources implement. Nodes involved in the hunting should all run the STX-Shifter to collect data. STIX-Shifter [177] is an open-source project by IBM that allows the software to connect with products that house data repositories by using STIX Patterning and return results as STIX Observations. Although Kestrel hunting language abstracts hunting knowledge codified in analytics and hunting flows, it does not automate the hunting process. In fact, an omniscient security analyst has to be involved in the hunting loop. Similarly, Microsoft Threat Intelligence introduced MSTICPY [178], a library for security investigation and reactive threat hunting using Jupyter Notebooks. MSTICPY is propped by a query language to (i) query log data from multiple sources, (ii) extract IoA from logs and unpack encoded data, (iii) perform analysis such as anomalous session detection and time series decomposition, and (iv) visualize data using interactive timelines, process trees, and multi-dimensional charts. Similar to other query languages (*e.g.,* Kestrel [176]), a security expert is presently needed to conduct the hunting by formulating those queries, analyzing data, and educing threat hypotheses.

[6]OSCF: www.github.com/ocsf
[7]OpenC2: www.openc2.org
[8]STIX-Shifter: www.github.com/opencybersecurityalliance/stix-shifter

C. Operational Languages

Trusted Automated eXchange of Indicator Information (TAXII) [179] is a cybersecurity protocol used to exchange CTI information over HTTPS using a client-server communication model. It has been designed particularly to support STIX information by defining an API that aligns with common sharing models. TAXII supports three main models: (i) *Hub and Spoke:* the hub acts as a repository of information while spokes can act as consumers or producers, (ii) *Source and Subscriber:* multiple subscribers utilize the same source of information, and (iii) *Peer-to-Peer:* multiple peer nodes share information using a P2P-like infrastructure. Security analysts can implement and combine different TAXII services such as (i) *Discovery:* to learn what services an entity supports and how to interact with them, (ii) *Collection Management:* to learn about and request subscriptions to data collections, (iii) *Inbox:* to receive content (*i.e.,* push messaging), and (iv) *Poll:* to request content (*i.e.,* pull messaging).

D. Sharing Platform

Malware Information Sharing Platform (MISP) [180] is an open-source threat intelligence platform initially sponsored by European Union. Even though not a standard, as MISP is widely used in the industry, we would comment on it in the following. MISP is exploited to develop effective utilities and documentation for threat intelligence, storing, correlating, and sharing IoC of targeted attacks, vulnerability, or even financial fraud and counter-terrorism information. To enable shared indicators and attributes among organizations, MISP uses different sighting standards and tools, such as STIX, for data exportation and representation [181].

E. Testbeds for Security and Analysis

Due to the huge technological adoption that leads to new protocols, a variety of applications, and massive generated data that makes security assessment and analysis challenging in run-time production environments; there is a need to develop comprehensive testbeds for security and analysis purposes [182]. Those testbeds not only help in testing new solutions and tools but also in providing timely assessment, better planning, and delivering technical analysis and reporting [183], [184]. Testbeds for security are environments (*e.g.,* virtual, hardware) set up specifically for testing and analyzing the security of software, networks, and other digital systems by simulating real-world scenarios and evaluating the effectiveness of security measures [185], [186].

Considering cybersecurity, digital twins can be used to simulate and predict cyber threats and vulnerabilities [187]. Indeed, digital twins can create a virtual replica of an organization's system and infrastructure and then test different security

scenarios. This process helps in identifying potential threats, breaches, and risks and thus proactively implementing security measures and defense lines. For instance, Faleiro et al. [188], explored the current state of digital twin research for cybersecurity by investigating its resilience. The authors covered different use cases including intrusion detection systems, simulation testing and training, privacy and legal compliance, and security for the factory of the future. Indeed, the digital twin becomes a repository for enormous data that could be collected from sensors that can be used to train security models to predict and provide suggestions based on the acquired information. In addition, security experts can develop different defense strategies and deploy them in a digital twin environment to validate them before using them in a runtime environment. The authors listed a few challenges that need to be addressed such as data privacy and human factor error that cannot be measured within a digital twin environment. Similarly, Pokhrel et al. [189] provided a review on the use of digital twins for predicting cybersecurity incidents. Indeed, the work identifies different use cases such as simulating cyber attacks, predicting vulnerabilities, and monitoring network behavior. Although the potential of digital twins to enhance security performance, developing accurate data and system models for digital twins is still a challenging task due to the complexity of the system, the huge amount of generated data, and the non-predictive model for the human factor. Pirbhulal et al. [190] designed a framework that involves the creation of a virtual replica of an IoT system using digital twin, which is used to simulate and predict potential threats and risks related to healthcare applications. The framework uses threat intelligence from physical works and then simulates different attacks in the digital twin field using traffic/attack generation and impact assessment. The authors proposed to apply security strategies to automate response and enforce the security of healthcare applications.

The work by NIST [191] presented a cybersecurity testbed design for Industrial Control Systems (ICS) and listed research use cases and performance metrics that need to be addressed. The testbed is composed of multiple control systems (*e.g.,* control of a chemical plant, dynamic assembly using robots, distributed supervision and control of large wide area networks, etc.). The objective of the testbed is to validate the existing security standards and provide guidance on the best practices for implementing cybersecurity strategies within an ICS by measuring different metrics (*e.g.,* process availability, quality, cost, etc.) and assessing the impact of security on the process performance. The study showed that security technologies had a direct impact on control performance, where security engineering is required to design control algorithms. Similarly, Green et al. [192] focused on the complex multidisciplinary challenge to design a cybersecurity testbed for ICS. The authors presented a non-prescriptive flexible high-level model – based on over six years of research on testbed and development, that is complemented by a set of practical guidelines related to emerging technologies. The high-level model is built on top of the user requirements and consists of a Remote Access Layer for external experimental platforms that communicate with Management, User, and Safety/Security Experimental Layer. The latter contains different IoT, SOC, and ICS, while the user layer comprises experimental devices, workstations, storage, and network/process collecting. The authors planned to extend the testbed to collect additional artifacts and data to be shared with the community to further validate ICS standards and security guidelines.

Testbeds for security and analysis using digital twins technology are critical for ensuring the security and reliability of digital systems by allowing security experts to identify vulnerabilities, breaches, and weaknesses before they can be exploited by attackers, and therefore develop effective security strategies to enforce the system's security.

F. Relation Between Industrial Security Tools & Threat Hunting

Before diving into threat hunting in the industry, it is important to understand the connection and distinction between SIEM, (X)SOAR, XDR, and their relation to threat hunting.

A SIEM [45] is a security software package that aims to provide reactive security by aggregating data from numerous systems, providing data analytics to detect abnormal behaviors, and generating alerts and responses if needed. SOAR and XSOAR tools [193] have been developed to improve SIEM platforms by (a) enabling organizations to collect inputs monitored by the security operations team, (b) simplifying the identification of critical incidents, (c) automating security operations and making them more efficient and effective, (d) allowing orchestration of different security technologies within one workflow, and (e) supporting the use of playbooks to automate responses to specific incidents. Indeed, XSOAR [194] is an improved version of SOAR that mainly focuses on specific features/capabilities and provides an interactive investigation for cross-team collaboration. On the other hand, XDR [195] is a unified security incident detection and response platform that incorporates a range of investigative tools such as behavioral analytics and automated remediation capabilities in order to speed up remediation and only escalate threats when a security expert intervention is required. An XDR tool centralizes and normalizes data from all connected sources, correlates all security data and alerts, and delivers a centralized incident detection and response capability with comprehensive monitoring across the entire attack surface. Contrastingly, threat hunting is an activity that strives to proactively discover stealthy threats and malicious activities before compromising the system. Security solutions, overwhelmingly, provide threat detection capabilities rather than pursuing advanced threats and adversaries. Due to advanced persistent threats and breaches, it is indispensable to extend the SIEM/SOAR/XDR solutions with threat hunting capabilities. The merger tends to enhance the security posture of organizations, improve investigative efficiency, determine the root cause of security incidents, and reduce time wasted on false positives.

SIEM and (X)SOAR solutions are popular tools that assist the SOC team to perform threat hunting by providing centralized visibility, analysis, and automated security processes. In the following, for the sake of space, we only present a few

tools and platforms that help in accelerating threat hunting. We believe these tools though represent the general characteristics of the majority of threat hunting tools and platforms.

- *Falcon* is a security platform developed by CrowdStrike. It comes with multiple tools such as Falcon Insight (*i.e.,* EDR), Falcon XDR, and Falcon OverWatch. Falcon Insight [196] monitors endpoint activities and analyzes the data in a real-time fashion to automatically identify threat activity and detect/prevent advanced threats as they happen using TTPs and IOAs. Falcon XDR [166] extends the EDR capabilities and delivers real-time multi-domain detection and orchestrated response to improve threat visibility across the enterprise, accelerate security operations, and reduce risk. Falcon OverWatch [197] is a managed threat hunting service. It conducts thorough human analysis to relentlessly hunt for anomalous or new attackers designed to evade other detection techniques. OverWatch uses threat intelligence to provide up-to-the-minute intelligence, applies complex statistical methods to examine outliers, and reconstructs the attack to comprehensively understand the attack and accelerate the investigation.
- *ArcSight ESM* [165] is a Micro Focus enterprise security management product that tends to provide (i) scalable event management (*e.g.,* data collection, aggregation, and normalization), (ii) real-time threat detection using correlation and customizable rule sets, and (iii) seamless integration with SOC tools, MITRE ATT&CK, and threat intelligence feed. ArcSight ESM helps in identifying and prioritizing security threats, organizing and tracking incident response activities, and simplifying audit and compliance activities by applying various big data security analytics and intelligence mechanisms.
- *Helix* [158] is a SIEM platform from FireEye that provides threat detection and alert management. It uses rule-based, signature-based, and non-signature-based mechanisms to detect and analyze advanced and non-malware-based threats. Helix comes with a managed rule-based system that can be integrated with threat intelligence to identify attacks using TTPs and IoCs. It also provides workflow and case management to generate investigation instructions with searches and actions to perform along with compliance and alert reporting.
- *Cisco XDR* [152] is a security solution that aims to improve the detection and response capabilities of Cisco's detection and response products. It allows organizations to collect, correlate, and analyze threat data as well as prioritize, hunt, and remediate risks. Cisco XDR security solution is part of the SecureX cloud-native platform, which integrates with all Cisco security products. SecureX is used to centralize security products and environments (*e.g.,* network security, cloud edge, and EDR) and provide a security orchestration platform that connects Cisco security services with infrastructure. SecureX provides different capabilities such as: (i) enabling unified visibility, including activity feeds, threat intelligence, and delivery of metrics; (ii) setting forth automated workflow and orchestration using a no or low code shareable playbooks; and (iii) providing managed threat hunting to hunt across domains while leveraging threat intelligence, threat story, and data correlation technologies; aiming at creating a consistent user experience, facilitating collaboration, and driving measurable insights.
- *Cortex XDR* [142] is an extended detection and response platform from Palo Alto Networks that aims to monitor and manage cloud, network, and endpoint events and data. Cortex XDR comes in two versions: (i) *Prevent* that provides protection for endpoints using an incident engine, integrated response capabilities, and threat intelligence feed; and (ii) *Pro* that appends more capabilities for networks, cloud resources, and third-party products such as behavior analytics, rule-based detection, accelerated investigation, and manual threat hunting. The latter requires careful searching by security experts through system and event data to identify indicators of compromise or behavioral indicators of compromise. Cortex XDR incorporates ML technologies to help detect known and unknown threats and XQL query language to query for the information contained in a wide variety of data sources. It also provides incident management, and automated root causes analysis.
- *QRadar* [143] is a SIEM solution from IBM that could be deployed as a hardware, software, or virtual appliance. QRadar includes: (i) event processors for collecting, storing, and analyzing event data; (ii) event collectors for capturing and forwarding data; (iii) flow processors for collecting Layer 4 network flows; and (iv) QFlow processors for performing deep packet inspection of Layer 7 application traffic. QRadar can also integrate modules for risk and vulnerability management, forensics analysis of packet captures, and incident response. It can support IBM X-Force Threat Intelligence [198] and other third-party threat intelligence feeds via STIX and TAXI to improve threat detection.
- *Splunk* [144] is a set of solutions that are used for searching, monitoring, visualizing, and analyzing machine data on a real-time basis. For instance, Splunk Enterprise is a system that collects and analyses the big data which is generated by the organization's infrastructure in order to get complete visibility of the security stack. Splunk Enterprise Security is a SIEM system that uses collected data from multiple sources to generate operational insights into threats, vulnerabilities, and security technologies. It also provides security analytics and threat detection and investigation capabilities using ML techniques. Splunk SOAR, on the other hand, tends to facilitate end-to-end security operations by orchestrating security workflows and automating tasks.

In general, these security tools accept a wide range of inputs (*e.g.,* CTI, logs, network traffic) and provide many capabilities for data collecting, processing, filtering, and correlating events. This is due to the fact that these tools are primarily SIEM or detection tools extended with threat hunting capabilities, therefore benefiting from mature techniques and approaches to security event collection and management techniques. These tools also generally provide workflows and

some automated tasks to respond to the attacks. They are sometimes coupled and tied into SOAR systems for further orchestration of responses and courses of action. In addition, many of the current solutions mention some level of analytics for threat hunting, though the absolute majority deploy rule-based techniques for threat hunting. Moreover, the output from the current solutions is mainly about helping the experts to manually search for the threats, present the attack story, and unified visibility into the threat landscape for the system. Overall, these solutions use a manual/humane involved approach for threat hunting. They currently do not present a high-level intention-based general framework to hunt threats and attacks.

VII. Summary & Discussion

Threat Detection vs Threat Hunting: The body of literature reveals a stark contrast between the academic attention given to threat detection and threat hunting. Numerous surveys, articles, and publications have been dedicated to exploring and expanding the efforts in threat detection [40]. Those efforts mainly focus on the reactive detection of existing attacks. Conversely, threat hunting – as a proactive defense line, has received markedly less focus from academia. However, industrial solutions proved that deploying proactive threat hunting mechanisms help strengthen security by retrieving stealthy attacks or attack campaigns that circumvent conventional threat detection mechanisms. In addition, our analysis showed that there are abundant studies and efforts on threat detection [199] that either (i) detect existing attacks by mining the past events captured in the organization's network [51] and/or (ii) predict the next step based on the previous activities and then detect the attack [200]. However, in threat hunting, there is a need to understand what happened in the past and compare it with what could happen as next steps or retrieving the traces on ongoing campaigns. Note that this is different from predicting the occurrence of future attacks, as predicting is not rooted in mixing the knowledge of the past and the knowledge about the attacks. In fact, proactive hunting persists in between a full understanding of past activities and an intelligent correlation for future steps based on a contextual understanding of what has been received as an IOC, which leads to generating relevant hypotheses for potential attacks [28], [29], [30].

Limitation of Rule-based and Statistical-based threat hunting: Based on our study, we found that there are fewer threat hunting research efforts using rule-based techniques (*e.g.,* [66], [105], [106], [107], [108]). This is due to the fact that rule-based solutions are (i) too permissive and straightforward, which might be easy to exploit and bypass; (ii) lack adaptability, which means they need constant effort to adapt to new types of attacks and therefore slow to adapt, costly and error-prone for today's dynamic threat landscape of adapting to new types of attacks; (iii) limited in scalability, which means difficult to create and maintain a comprehensive set of rules. On the other hand, existing statistical techniques (*e.g.,* [103], [104], [163], [164]) do not have the capabilities to take into account contextual information. In fact, capturing the context of ongoing attacks in a dynamic run-time environment is challenging and might result in: (i) a high level of false-positive alerts as any deviation from normal behavior can trigger an alert, (ii) difficulties in dealing with low and slow attacks as the long-period behavior cannot be captured and distinguished from normal behaviors, and (iii) challenges in defining the normal and accurate baseline, especially for stealthy attacks that are difficult to be captured.

Graph-Based Promises: We also found that there is, in recent years, an increasing number of adoption of graph-based solutions [20], [109], [110], [111], [112], [113] in threat hunting. The reason behind this is that graph-based solutions [201] are particularly good at: (i) capturing and preserving the semantics and relationships between different entities, interactions, and actions; (ii) providing relationship-based analysis by emphasizing the relationships and their semantics; (iii) understanding the graph structure to uncover new attack patterns; and (iv) providing recommendation on effective countermeasures. Therefore they clearly alleviate some of the shortcomings when it comes to rule-based and statistical approaches used for TH.

ML/AI as a New Trend: The study also showed that there is a big shift towards the adoption of ML/AI techniques (*e.g.,* [121], [122], [123], [124], [125], [126], [127], [128], [129], [130]). This is due to the fact that ML-based solutions can dynamically learn from historical data [202] without relying on rules-based programming or formalizing relationships between variables [203], In addition, ML-based solutions can provide self-learning capabilities over time based on the change in the data and users' behaviors (*e.g.,* change in the attacker's behavior), which in return helps in reducing the time and efforts required by SOC team [204]. However, the quality and quantity of data required to be processed for threat hunting can be challenging to collect and retain [205]. There must be sufficient data collected, from a sufficient number of data sources and locations within an environment, to enable balanced intelligent analysis. However, what is sufficient can vary greatly based on context and target domains and is often unknowable in advance, making this type of hunting hard to utilize effectively. Overall, ML techniques can be employed to efficiently map different attack techniques to different APTs, which in return helps in providing threat actor classification. This classification can be used by SOC teams to prioritize attacks based on their importance/severity (*e.g.,* national states attack, organized crime, stand-alone insiders). ML/AI techniques, due to their capabilities and the increasing availability of data, have been a focal point of research on threat hunting in recent years. For all the reasons mentioned before, they seem to be promising for efficient threat hunting. In addition, automating the hunting process (*e.g.,* [101], [118], [119], [120]) helps in addressing attack fatigue by requiring less intervention of security analysis.

MITRE ATT&CK relevance: Given the increasing nature of the attacks and attack campaigns, the solutions target more complex attack campaigns often based on the MITRE ATT&CK framework (*e.g.,* [71], [76], [161], [186], [206]). This framework has become a de facto standard for documenting adversary behavior and facilitating its discovery and detection. Indeed, the MITRE ATT&CK enhances the

cybersecurity operations from threat detection and hunting to incident response and defense optimization by providing the following capabilities: (i) comprehensive understanding of attack techniques that helps security analysts to understand the threat landscape and various strategies an attacker might use; (ii) use of a common language and representation to represent cyber threat behaviors that eventually helps in effective communication, collaboration, and sharing between various organizations and security bodies; (iii) prioritizing the defense by identifying the most relevant techniques for each attack step and security gaps; and (iv) evaluating security strategies by providing an unbiased assessment on how these strategies can detect and hunt attacks.

VIII. Lessons Learned

Through our study, we observed that many existing approaches share critical limitations toward a successful hunting operation. In the following, we outline a few limitations, research challenges, and potential further directions.

A. Automated Hypothesis Generation and Verification

Limitations and Challenges: The threat hunting process is initialized by the generation of hypotheses. These hypotheses are often hand-crafted manually by security analysts tending to prove the existence of an attack/threat. The generation process requires not only knowledge and expertise, which are elusive, but also costly in time and computing to analyze a huge amount of input data. The hypotheses are then verified and validated to confirm or refute the existence of threats, which might lead to the discovery of new TTP instances [207]. In modern Enterprise networks, the immense number of applications, users, and data, in addition to massive virtualization and cloudification of networking, storage and computation, entail the need to automate the hunting process.

Research Directions: Hypotheses – as they are the starting point of hunting, should be generated automatically on the fly based on the current system status and available data. Graph theory approaches and ML techniques could be leveraged to generate quality hypotheses. The graph will be used to capture and represent the semantics of hypothesis items (context, involved techniques, and targeted threats) as well as perform pruning to preserve only the critical and the most likely hypotheses. Similarly, ML techniques can also be used to predict the attacker's next step and hence generate potential threat hypotheses based on the attacker's expected moves. The generated hypotheses could be verified and validated using an automated reasoner without the need for human intervention. Indeed, machine reasoning can be implemented as pillar elements to check each hypothesis, collect additional data (if needed), and then perform backward/forward reasoning to provide the existence of a threat. Here, the softwarization in the modern virtualized environment may be instrumental to creating run-time dynamic verification agents or security appliances (*e.g.,* creation of targeted collectors, EDR agents, or security appliances in different virtual networks and virtual machines to better verify or hunt down the possible threats). Although automating hypothesis generation and verification help in avoiding the high percentage of false-positive alerts with satisfying precision/accuracy, it has been proved difficult so far due to the complexity of the inference engine and the tremendous knowledge space.

B. Intelligent Data Representation

Limitations and Challenges: In light of the previous point, the massive amount of available/generated data triggers three main challenges: (i) identifying what data is relevant in regard to the hunting performance and model training/learning, (ii) designing optimized storage techniques, and (iii) introducing efficient data representation to facilitate the analysis, retrieval, and transmission. Intelligent data representation schemes are highly required to facilitate the hunting process by representing actors, relationships, and context. In fact, similar efforts started to take shape in other areas such as STIX [133] and TAXII [179]. Thus, there is a need to extend these efforts to data representation in order to have a uniform representation of different systems and hence facilitate the use of threat hunting tools in different environments. Knowledge graphs (*e.g.,* Neo4j, Azure Cosmos DB, Amazon Neptune) are mostly suitable for this tasks [208], yet, the huge density of data and their relationships could conduct in an explosion in the size graph, which in turn leads to performance degradation [209].

Research Directions: To tackle this issue, it becomes necessary to append cleaning and context/relevant-driven optimization techniques to the knowledge graph aiming at avoiding the size explosion while respecting the semantic context and relations [208]. These techniques are not yet adapted due to the nature and format of input data in cybersecurity, which is challenging to be unified. Although using knowledge graphs helps in contextual understanding and information sharing that helps threat hunting, it also presents challenges related to data integration (due to the diversity of resources and formats), scalability (due to the massive amount of generated data), maintenance (due to the rapidity evolve of cybersecurity landscape), and interpretation (due to the massive number of relationships among nodes). Moreover, an efficient data representation should also help in facilitating hunting rules generation without the need for experts' inputs. In fact, the generated hunting rules must match the data representation and approbate and harmonize with the logical/semantic relations within the organization's system. Yet, those rules need to be validated by an automated system and security expert to ensure they will not block legitimate activities or lead to false positives.

C. Intelligent Security Awareness

Limitations and Challenges: Current security systems do not adapt to dynamic changes/situations (*e.g.,* external environmental factors) and must escalate to analysts' support for better decision-making [210]. Decisions within policy-based automation are made at design time coupled between use cases, policies, and underlying infrastructure. Event-condition-action (ECA)-based policy automation (*e.g.,* security playbooks, firewall rules, etc.) is made at design time coupled between use cases and underlying infrastructure.

Research Directions: To assist SOC team with run-time intelligent decisions based on the changing security threats and environmental factors, the security engine has to decouple the "*conditions*" from the "*actions*." One way to enforce security awareness is to design intent-based security [211] that establishes machine-readable knowledge about goals, targets, requirements, and constraints. Intent-based security requires a combination of user behavior analytics platforms, threat intelligence tools, and automation systems, as well as a strong understanding of organizational context and risk profiles. The intent data is then fed to the threat hunting engine to proactively identify potential threats and vulnerabilities. This data can include IoC, known attack patterns, and other contextual information. By incorporating intent data into threat hunting activities, security teams can proactively search for indicators or behavioral patterns that align with known malicious activities and perform automation and orchestration of different actions based on the security context and policy. This approach enables more targeted and effective threat hunting.

D. Quantity & Quality of Data

Limitations and Challenges: The combination of multiple protocols in modern enterprise networks and the increasing number of users and IoT devices/sensors leads to generating a huge amount of data with different structures, various formats, and disparate schema [212]. In addition, APIs and service function chaining in modern networks generate flow and log information based on the application/use case rather than the network level. This massive data carries information about benign and malicious/suspicious nodes and activities and will be used as input to train threat hunting models (*e.g.,* statistical, machine learning).

Research Directions: Although the quantity of data is important to training the model, the quality of data is also important in order to generate an accurate and unbiased model [213]. In active learning, fewer yet quality data can help in training the model faster with high accuracy. In most organizations, collected data is unbalanced where threats and abnormal behaviors have a small portion. This will lead to generating a biased model that skews towards normal behaviors. To overcome this issue, the following research directions should be taken into consideration: (a) *diverse and representative training data:* it is crucial to ensure diverse and comprehensive training datasets that include a wide range of attack scenarios, including rare and emerging threats; (b) *data sampling:* it is important to use over-sampling and under-sampling to balance the normal and abnormal data samples in the training dataset; (c) *feature engineering and selection:* it is important to carefully select features that capture both normal and anomalous behavior by considering incorporating domain knowledge and expert insights to identify relevant features that represent rare attacks; (d) *continuous model evaluation and retraining:* regular evaluation and retraining of models are crucial to identify and addressing bias (particularly for rare attacks) and using the feedback to update and improve the model; (e) *continuous threat intelligence integration:* it is important to stay updated with the latest threat intelligence to ensure that the model is trained on current attack trends and patterns (valuable information about new attack vectors, zero-day exploits, and emerging threats); and (f) *human-in-the-loop approach:* security expertise is invaluable in detecting and understanding rare attacks that eventually help in improving the model accuracy.

E. Attacks Assisted by AI

Limitations and Challenges: AI-based tools (*e.g.,* OpenAI ChatGBT,[9] Google Bard[10]) [214] might pose significant risks when used to generate new cyberattacks and threats. Those tools can be harmfully used to generate more convincing phishing emails, create deep-fake audio/video content, automate vulnerability scanning, manage and coordinate botnets, etc. Conventional security solutions might not be sophisticated to defend against those tools due to the speed, scale, lower barrier to entry, and erosion of trust those tools provide.

Research Directions: Hunting AI-assisted cybersecurity attacks requires a combination of traditional and AI-specific security measures, such as: (i) implementing security threat hunting mechanisms that specifically target AI-assisted attacks such as adversarial training, (ii) developing proactive plans involving AI/ML techniques for adapting the hunting mechanisms to such attacks, and (iii) better information exchange with industry peers, agencies, and stakeholders to share existing AI-assisted attacks and best practices for their hunting.

F. Integration With (X)SOAR

Limitations and Challenges: The ultimate objective of security analysts is not only to discover new threats in the system but also to produce a course of action to eliminate threats [215]. Based on the results of hypotheses validation, an efficient threat hunting system needs to be tightly integrated with SOAR in order to automatically recommend a response and workflow based on the threat and its level of severity (*e.g.,* emerging NFV-MANO in Telecom network for management and orchestration).

Research Directions: The input/output formats and information exchange should be standardized to allow smooth and more efficient communications and CTI dissemination among different security components and collaborators. For example, IBM Kestrel [176] addresses this issue by using multiple data sources collected via the Telemetry & Threat Intel Data components (EDR, NDR, SIEM, TI, etc.)[11] and then reinforced using a domain-specific query language, though this could be avoided if a unique standard data representation could have been used. This will also make it easier to use/compare different threat hunting tools for the same environment. This integration helps in designing an automated-closed hunting loop, starting from the hypotheses generation, uncovering new threats, to automating the recommended course of action. Indeed, threat hunting should be working in line with (X)SOAR through different internal APIs to provide proactive insights about existing threats

[9]OpenAI ChatGPT: https://chat.openai.com/
[10]Google Bard: https://bard.google.com/
[11]What is Kestrel?: https://kestrel.readthedocs.io/

and help the automation and orchestration of the course of actions that need to be executed to stop the damage before it happens.

IX. Conclusion

Modern cyberattacks are advanced to avoid the defenses offered by conventional security measures (*e.g.,* threat detection). Attackers become more skilled and deploy cutting-edge techniques that lead to frequent and sophisticated attacks. These advanced threats persist and could be unnoticed for months in the network. Unlike reactive security strategies such as automated threat detection systems, threat hunting is a forward-looking security approach in which threat hunters investigate myriad and tedious log and user activities to uncover undetectable stealthy attacks and suspicious activities. In this paper, we focused on threat hunting at an Enterprise network level. More precisely, we studied the threat hunting concept, and we designed a unified top-down architecture for threat hunting. To the best of our knowledge, this is the first systematic and detailed component-based architecture of threat hunting. In addition, we comprehensively surveyed various solutions proposed to hunt threats/attacks, provided a taxonomy based on the used techniques such as machine learning, graph-based solutions, rule-based, and statistical-based methods, and further classified these solutions based on the detailed approach. Moreover, we discussed the existing standardization efforts (*e.g.,* representation, query languages, operation languages, and sharing) as well as testbed and relation with the industry. In the end, we also presented a discussion based on our analysis and identified various research gaps and challenges that may be considered by the research community to design concrete and efficient threat hunting solutions.

Acknowledgment

The authors would like to thank the anonymous reviewers for their constructive comments and suggestions that helped them to further improve the quality, content, and presentation of this paper.

References

[1] *State of Cybersecurity Resilience 2021: How Aligning Security and the Business Creates Cyber Resilience*, Accenture, Dublin, Ireland, 2021.
[2] *Cybersecurity Almanac: 100 Facts, Figures, Predictions and Statistics*, Cisco/Cybersecurity Ventures, San Jose, CA, USA, 2020.
[3] L. Shen, "The NIST cybersecurity framework: Overview and potential impacts," *Scitech Lawyer*, vol. 10, no. 4, p. 16, 2014.
[4] W. U. Hassan, A. Bates, and D. Marino, "Tactical provenance analysis for endpoint detection and response systems," in *Proc. IEEE Symp. Security Privacy (SP)*, 2020, pp. 1172–1189.
[5] N. Capuano, G. Fenza, V. Loia, and C. Stanzione, "Explainable artificial intelligence in cybersecurity: A survey," *IEEE Access*, vol. 10, pp. 93575–93600, 2022.
[6] A. Oktadika, C. Lim, and K. Erlangga, "Hunting cyber threats in the enterprise using network defense log," in *Proc. IEEE Int. Conf. Inf. Commun. Technol. (ICoICT)*, 2021, pp. 528–533.
[7] A. Valdes and K. Skinner, "Adaptive, model-based monitoring for cyber attack detection," in *Proc. Int. Workshop Recent Adv. Intrusion Detection*, 2000, pp. 80–93.
[8] O. Niggemann, S. Windmann, S. Volgmann, A. Bunte, and B. Stein, "Using learned models for the root cause analysis of cyber-physical production systems," in *Proc. Int. Workshop Principle Diagn. (DX)*, 2014, pp. 1–9.
[9] X. Shu et al., "Threat intelligence computing," in *Proc. ACM Conf. Comput. Commun. Security (SIGSAC)*, 2018, pp. 1883–1898.
[10] A. Bhardwaj and S. Goundar, "A framework for effective threat hunting," *Netw. Security*, vol. 2019, no. 6, pp. 15–19, 2019.
[11] Z. Wang, "A systematic literature review on cyber threat hunting," 2022, *arXiv:2212.05310*.
[12] S. Mansfield-Devine, "Threat hunting: Assuming the worst to strengthen resilience," *Netw. Security*, vol. 2017, no. 5, pp. 13–17, 2017.
[13] Q. Liu et al., "Latte: Large-scale lateral movement detection," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, 2018, pp. 1–6.
[14] B. Bowman, C. Laprade, Y. Ji, and H. H. Huang, "Detecting lateral movement in enterprise computer networks with unsupervised graph AI," in *Proc. Int. Symp. Res. Attacks Intrusions Defenses (RAID)*, 2020, pp. 257–268.
[15] M. Thangavelu, V. Krishnaswamy, and M. Sharma, "Impact of comprehensive information security awareness and cognitive characteristics on security incident management—An empirical study," *Comput. Security*, vol. 109, Oct. 2021, Art. no. 102401.
[16] T. Madi, H. A. Alameddine, M. Pourzandi, and A. Boukhtouta, "NFV security survey in 5G networks: A three-dimensional threat taxonomy," *Comput. Netw.*, vol. 197, Oct. 2021, Art. no. 108288.
[17] D. Schlette, M. Caselli, and G. Pernul, "A comparative study on cyber threat intelligence: The security incident response perspective," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 4, pp. 2525–2556, 4th Quart., 2021.
[18] "A framework for cyber threat hunting," Sqrrl Data, Gurgaon, Haryana, Rep. 9, 2018. [Online]. Available: https://www.threathunting.net/files/framework-for-threat-hunting-whitepaper.pdf
[19] R. Ruefle, A. Dorofee, D. Mundie, A. D. Householder, M. Murray, and S. J. Perl, "Computer security incident response team development and evolution," *IEEE Security Privacy*, vol. 12, no. 5, pp. 16–26, Sep./Oct. 2014.
[20] F. K. Kaiser et al., "Attack hypotheses generation based on threat intelligence knowledge graph," *IEEE Trans. Depend. Secure Comput.*, early access, Jan. 4, 2023, doi: 10.1109/TDSC.2022.3233703.
[21] I. Kotenko, K. Izrailov, and M. Buinevich, "Static analysis of information systems for IoT cyber security: A survey of machine learning approaches," *Sensors*, vol. 22, no. 4, p. 1335, 2022.
[22] S. Majumdar et al., "ProSAS: Proactive security auditing system for clouds," *IEEE Trans. Depend. Secure Comput.*, vol. 19, no. 4, pp. 2517–2534, Jul./Aug. 2022.
[23] H. Kermabon-Bobinnec et al., "ProSPEC: Proactive security policy enforcement for containers," in *Proc. ACM Conf. Data Appl. Security Privacy (CODASPY)*, 2022, pp. 155–166.
[24] W. Zhijun, L. Wenjing, L. Liang, and Y. Meng, "Low-rate DoS attacks, detection, defense, and challenges: A survey," *IEEE Access*, vol. 8, pp. 43920–43943, 2020.
[25] W. Li, W. Meng, and L. F. Kwok, "Surveying trust-based collaborative intrusion detection: State-of-the-art, challenges and future directions," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 280–305, 1st Quart., 2021.
[26] T. Mahjabin, Y. Xiao, G. Sun, and W. Jiang, "A survey of distributed denial-of-service attack, prevention, and mitigation techniques," *Int. J. Distrib. Sensor Netw.*, vol. 13, no. 12, p. 29, 2017.
[27] V. D. M. Rios, P. R. Inacio, D. Magoni, and M. M. Freire, "Detection and mitigation of low-rate denial-of-service attacks: A survey," *IEEE Access*, vol. 10, pp. 76648–76668, 2022.
[28] R. M. Lee and R. T. Lee, *SANS 2018 Threat Hunting Survey Results*, SANS Inst., Rockville, VA, USA, 2018.
[29] M. Fuchs and J. Lemon, *SANS 2019 Threat Hunting Survey: The Differing Needs of New and Experienced Hunters*, SANS Inst., Rockville, VA, USA, 2019.
[30] R. Brown and R. M. Lee, *2021 SANS Cyber Threat Intelligence (CTI) Survey*, SANS Inst., Rockville, VA, USA, 2021.
[31] F. Aldauiji, O. Batarfi, and M. Bayousif, "Utilizing cyber threat hunting techniques to find ransomware attacks: A survey of the state of the art," *IEEE Access*, vol. 10, pp. 61695–61706, 2022.
[32] S. Homayoun et al., "DRTHIS: Deep ransomware threat hunting and intelligence system at the fog layer," *Future Gener. Comput. Syst.*, vol. 90, pp. 94–104, Jan. 2019.
[33] Q. Wang et al., "You are what you do: Hunting stealthy malware via data provenance analysis," in *Proc. Netw. Distrib. Syst. Security (NDSS)*, 2020, pp. 1–8.
[34] T. Komárek, J. Brabec, Č. Škarda, and P. Somol, "Threat hunting as a similarity search problem on multi-positive and unlabeled data," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, 2021, pp. 2098–2103.

[35] J. Li et al., "LogKernel: A threat hunting approach based on behaviour provenance graph and graph kernel clustering," *Security Commun. Netw.*, vol. 2022, Sep. 2022, Art. no. 4577141.
[36] F. Yang, Y. Han, Y. Ding, Q. Tan, and Z. Xu, "A flexible approach for cyber threat hunting based on kernel audit records," *Cybersecurity*, vol. 5, no. 1, p. 11, 2022.
[37] S. M. Milajerdi, B. Eshete, R. Gjomemo, and V. Venkatakrishnan, "POIROT: Aligning attack behavior with kernel audit records for cyber threat hunting," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2019, pp. 1795–1812.
[38] M. Mahmoud, M. Mannan, and A. Youssef, "APTHunter: Detecting advanced persistent threats in early stages," *Digit. Threats Res. Pract.*, vol. 4, no. 1, p. 31, 2022.
[39] W. Xiong and R. Lagerström, "Threat modeling—A systematic literature review," *Comput. Security*, vol. 84, pp. 53–69, Jul. 2019.
[40] A. D. Raju, I. Y. Abualhaol, R. S. Giagone, Y. Zhou, and S. Huang, "A survey on cross-architectural IoT malware threat hunting," *IEEE Access*, vol. 9, pp. 91686–91709, 2021.
[41] W. Wideł, M. Audinot, B. Fila, and S. Pinchinat, "Beyond 2014: Formal methods for attack tree-based security modeling," *ACM Comput. Surveys*, vol. 52, no. 4, pp. 1–36, 2019.
[42] Z. Li, Q. A. Chen, R. Yang, Y. Chen, and W. Ruan, "Threat detection and investigation with system-level provenance graphs: A survey," *Comput. Security*, vol. 106, Jul. 2021, Art. no. 102282.
[43] L. Chen, R. Jiang, C. Lin, and A. Li, "A survey on threat hunting: Approaches and applications," in *Proc. IEEE Int. Conf. Data Sci. Cyberspace (DSC)*, 2022, pp. 340–344.
[44] J. Hou, Y. Li, J. Yu, and W. Shi, "A survey on digital forensics in Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 1–15, Jan. 2020.
[45] G. González-Granadillo, S. González-Zarzosa, and R. Diaz, "Security information and event management (SIEM): Analysis, trends, and usage in critical infrastructures," *Sensors*, vol. 21, no. 14, p. 4759, 2021.
[46] J. C. Sancho, A. Caro, M. Ávila, and A. Bravo, "New approach for threat classification and security risk estimations based on security event management," *Future Gener. Comput. Syst.*, vol. 113, pp. 488–505, Dec. 2020.
[47] S. Torabi, E. Bou-Harb, C. Assi, and M. Debbabi, "A scalable platform for enabling the forensic investigation of exploited IoT devices and their generated unsolicited activities," *Forensic Sci. Int. Digit. Invest.*, vol. 32, Apr. 2020, Art. no. 300922.
[48] M. Husák, J. Komárková, E. Bou-Harb, and P. Čeleda, "Survey of attack projection, prediction, and forecasting in cyber security," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 640–660, 1st Quart., 2018.
[49] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "Adversarial attacks and defences: A survey," 2018, *arXiv:1810.00069*.
[50] S. Sengupta, A. Chowdhary, A. Sabur, A. Alshamrani, D. Huang, and S. Kambhampati, "A survey of moving target defenses for network security," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1909–1941, 3rd Quart., 2020.
[51] A. Alshamrani, S. Myneni, A. Chowdhary, and D. Huang, "A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1851–1877, 2nd Quart., 2019.
[52] F. Salahdine and N. Kaabouch, "Security threats, detection, and countermeasures for physical layer in cognitive radio networks: A survey," *Phys. Commun.*, vol. 39, Apr. 2020, Art. no. 101001.
[53] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network intrusion detection for IoT security based on learning techniques," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2671–2701, 3rd Quart., 2019.
[54] N. Shinde and P. Kulkarni, "Cyber incident response and planning: A flexible approach," *Comput. Fraud Security*, vol. 2021, no. 1, pp. 14–19, 2021.
[55] N. H. A. Rahman and K.-K. R. Choo, "A survey of information security incident handling in the cloud," *Comput. Security*, vol. 49, pp. 45–69, Mar. 2015.
[56] P. Vadrevu and R. Perdisci, "What you see is NOT what you get: Discovering and tracking social engineering attack campaigns," in *Proc. Internet Meas. Conf.*, 2019, pp. 308–321.
[57] P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, "A detailed investigation and analysis of using machine learning techniques for intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 686–728, 1st Quart., 2018.
[58] A. Villalón-Huerta, I. Ripoll-Ripoll, and H. Marco-Gisbert, "A taxonomy for threat actors' delivery techniques," *Appl. Sci.*, vol. 12, no. 8, p. 3929, 2022.
[59] V. Mavroeidis, R. Hohimer, T. Casey, and A. Jesang, "Threat actor type inference and characterization within cyber threat intelligence," in *Proc. IEEE Int. Conf. Cyber Conflict (CyCon)*, 2021, pp. 327–352.
[60] *Cybersecurity Incident & Vulnerability Response playbooks—Operational Procedures for Planning and Conducting Cybersecurity Incident and Vulnerability Response Activities in FCEB Information Systems*, Cybersecurity Infrastruct. Security Agency, Rosslyn, VA, USA, Nov. 2021.
[61] M. D. Bruijne, M. V. Eeten, C. H. Ga˜ñán, and W. Pieters, "Towards a new cyber threat actor typology," Fac. Technol., Policy Manage., Delft Univ. Technol., Delft, The Netherlands, document WODC Rapport 2740, 2017, doi: 20.500.12832/2299.
[62] C. Colwill, "Human factors in information security: The insider threat—Who can you trust these days?" *Inf. Security*, vol. 14, no. 4, pp. 186–196, 2009.
[63] *Global Threat Report*, Crowdstrike, Sunnyvale, CA, USA, 2023.
[64] J. Hunker and C. W. Probst, "Insiders and insider threats—An overview of definitions and mitigation techniques," *J. Wireless Mobile Netw. Ubiquitous Comput. Depend. Appl.*, vol. 2, no. 1, pp. 4–27, 2011.
[65] L. Qiang, Y. Zeming, L. Baoxu, J. Zhengwei, and Y. Jian, "Framework of cyber attack attribution based on threat intelligence," in *Proc. 2nd Int. Conf. Interoper. Safety Security IoT (InterIoT)*, 2017, pp. 92–103.
[66] V. Mavroeidis and A. Jøsang, "Data-driven threat hunting using sysmon," in *Proc. Int. Conf. Cryptography Security Privacy (CSP)*, 2018, pp. 82–88.
[67] V. Pourahmadi, H. A. Alameddine, M. A. Salahuddin, and R. Boutaba, "Spotting anomalies at the edge: Outlier exposure-based cross-silo federated learning for DDoS detection," *IEEE Trans. Depend. Secure Comput.*, early access, Nov. 25, 2022, doi: 10.1109/TDSC.2022.3224896.
[68] N. Lukova-Chuiko, A. Fesenko, H. Papirna, and S. Gnatyuk, "Threat hunting as a method of protection against cyber threats," in *Proc. IT I*, 2020, pp. 103–113.
[69] S. Majumdar et al., "Multi-level proactive security auditing for clouds," in *Proc. IEEE Conf. Depend. Secure Comput. (DSC)*, 2019, pp. 1–8.
[70] Y. Kazato, Y. Nakagawa, and Y. Nakatani, "Improving maliciousness estimation of indicator of compromise using graph convolutional networks," in *Proc. IEEE 17th Annu. Consum. Commun. Netw. Conf. (CCNC)*, 2020, pp. 1–7.
[71] S. Bagui et al., "Detecting reconnaissance and discovery tactics from the MITRE ATT&CK framework in Zeek conn logs using spark's machine learning in the big data framework," *Sensors*, vol. 22, no. 20, p. 7999, 2022.
[72] T. Madi, H. A. Alameddine, M. Pourzandi, A. Boukhtouta, M. Shoukry, and C. Assi, "AutoGuard: A dual intelligence proactive anomaly detection at application-layer in 5G networks," in *Proc. Eur. Symp. Res. Comput. Security (ESORICS)*, 2021, pp. 715–735.
[73] "What is threat hunting?" Accessed: Sep. 1, 2022. [Online]. Available: https://www.ibm.com/topics/threat-hunting
[74] F. Maymí, R. Bixler, R. Jones, and S. Lathrop, "Towards a definition of cyberspace tactics, techniques and procedures," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, 2017, pp. 4674–4679.
[75] E. M. Hutchins et al., "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," *Leading Issues Inf. Warfare Security Res.*, vol. 1, no. 1, p. 80, 2011.
[76] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, *MITRE ATT&CK: Design and Philosophy*, MITRE, McLean, VA, USA, 2018.
[77] Z. Long, L. Tan, S. Zhou, C. He, and X. Liu, "Collecting indicators of compromise from unstructured text of cybersecurity articles using neural-based sequence labeling," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, 2019, pp. 1–8.
[78] M. Husák, T. Jirsík, and S. J. Yang, "SoK: Contemporary issues and challenges to enable cyber situational awareness for network security," in *Proc. 15th Int. Conf. Availability Rel. Security*, 2020, pp. 1–10.
[79] R. M. Lee and R. Lee, *The Who, What, Where, When, Why and how of Effective Threat Hunting*, SANS Inst., Rockville, VA, USA, 2016. [Online]. Available: https://www.sans.org/white-papers/who-what-where-when-why-how-effective-threat-hunting/
[80] T. Liliengren and P. Löwenadler. "Threat hunting, definition and framework." 2018. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:hh:diva-36759

[81] K. Wafula and Y. Wang, "CARVE: A scientific method-based threat hunting hypothesis development model," in *Proc. IEEE Int. Conf. Electron. Inf. Technol. (EIT)*, 2019, pp. 1–6.

[82] B. Fouss, D. Ross, S. Robinson, and K. Alperin, "NetSet: A set visualization tool for network metadata exploration and threat hunting." B. Fouss, D. Ross, S. Robinson, and K. Alperin. "NetSet: A set visualization tool for network metadata exploration and threat hunting." 2018. [Online]. Availbale: https://vizsec.org/files/2018/Fouss_Poster.pdf

[83] A. Oqaily, Y. Jarraya, L. Wang, M. Pourzandi, and S. Majumdar, "MLFM: Machine learning meets formal method for faster identification of security breaches in network functions virtualization (NFV)," in *Proc. Eur. Symp. Res. Comput. Security*, 2022, pp. 466–489.

[84] I. Yun, D. Kapil, and T. Kim, "Automatic techniques to systematically discover new heap exploitation primitives," in *Proc. USENIX Security Symp. (USENIX Security)*, 2020, pp. 1111–1128.

[85] M. R. Fatemi and A. A. Ghorbani, "Threat hunting in windows using big security log data," in *Security, Privacy, and Forensics Issues in Big Data*. Hershey, PA, USA: IGI Global, 2020, pp. 168–188.

[86] K. A. Akbar, S. M. Halim, Y. Hu, A. Singhal, L. Khan, and B. Thuraisingham, "Knowledge mining in cybersecurity: From attack to defense," in *Proc. Annu. IFIP WG Conf. Data Appl. Security Privacy XXXVI (DBSec)*, 2022, pp. 110–122.

[87] T. D. Wagner, K. Mahbub, E. Palomar, and A. E. Abdallah, "Cyber threat intelligence sharing: Survey and research directions," *Comput. Security*, vol. 87, Nov. 2019, Art. no. 101589.

[88] M. T. Alam, D. Bhusal, Y. Park, and N. Rastogi, "Looking beyond IoCs: Automatically extracting attack patterns from external CTI," 2022, *arXiv:2211.01753*.

[89] D. W. Baker, S. M. Christey, W. H. Hill, and D. E. Mann, "The development of a common enumeration of vulnerabilities and exposures," in *Proc. Recent Adv. Intrusion Detect.*, vol. 7, 1999, p. 9.

[90] M. Walkowski, J. Oko, and S. Sujecki, "Vulnerability management models using a common vulnerability scoring system," *Appl. Sci.*, vol. 11, no. 18, p. 8735, 2021.

[91] X. Jing, Z. Yan, and W. Pedrycz, "Security data collection and data analytics in the Internet: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 586–618, 1st Quart., 2018.

[92] J.-H. Cho et al., "Toward proactive, adaptive defense: A survey on moving target defense," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 709–745, 1st Quart., 2020.

[93] N. Sun, J. Zhang, P. Rimba, S. Gao, L. Y. Zhang, and Y. Xiang, "Data-driven cybersecurity incident prediction: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1744–1772, 2nd Quart., 2018.

[94] T. Taylor, F. Araujo, and X. Shu, "Towards an open format for scalable system telemetry," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, 2020, pp. 1031–1040.

[95] M. Usman, M. A. Jan, X. He, and J. Chen, "A survey on representation learning efforts in cybersecurity domain," *ACM Comput. Surveys*, vol. 52, no. 6, pp. 1–28, 2019.

[96] Y. Xin et al., "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, 2018.

[97] H. Al-Mohannadi, Q. Mirza, A. Namanya, I. Awan, A. Cullen, and J. Disso, "Cyber-attack modeling analysis techniques: An overview," in *Proc. IEEE Int. Conf. Future Internet Things Cloud Workshops (FiCloudW)*, 2016, pp. 69–76.

[98] E. Hemberg et al., "Linking threat tactics, techniques, and patterns with defensive weaknesses, vulnerabilities and affected platform configurations for cyber hunting," 2020, *arXiv:2010.00533*.

[99] A. Adedoyin and H. Teymourlouei, "Methods for automating threat hunting and response," in *Proc. Int. Conf. Elect. Comput. Energy Technol. (ICECET)*, 2021, pp. 1–6.

[100] C. Sayan, S. Hariri, and G. Ball, "Cyber security assistant: Design overview," in *Proc. IEEE Int. Workshops Found. Appl. Self Syst. (FAS W)*, 2017, pp. 313–317.

[101] F. Araujo, D. Kirat, X. Shu, T. Taylor, and J. Jang, "Evidential cyber threat hunting," 2021, *arXiv:2104.10319*.

[102] K. Thakur, M. Qiu, K. Gai, and M. L. Ali, "An investigation on cyber security threats and security models," in *Proc. IEEE 2nd Int. Conf. Cyber Security Cloud Comput.*, 2015, pp. 307–311.

[103] H. Almohannadi, I. Awan, J. A. Hamar, A. Cullen, J. P. Disso, and L. Armitage, "Cyber threat intelligence from honeypot data using elasticsearch," in *Proc. IEEE Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, 2018, pp. 900–906.

[104] A. B. Ajmal, M. A. Shah, C. Maple, M. N. Asghar, and S. U. Islam, "Offensive security: Towards proactive threat hunting via adversary emulation," *IEEE Access*, vol. 9, pp. 126023–126033, 2021.

[105] Y. Shen and G. Stringhini, "ATTACK2VEC: Leveraging temporal word embeddings to understand the evolution of cyberattacks," in *Proc. USENIX Security Symp. (USENIX Security)*, 2019, pp. 905–921.

[106] G. Ho et al., "Hopper: Modeling and detecting lateral movement," in *Proc. USENIX Security Symp. (USENIX Security)*, 2021, pp. 3093–3110.

[107] P. Radoglou-Grammatikis et al., "TRUSTY: A solution for threat hunting using data analysis in critical infrastructures," in *Proc. IEEE Int. Conf. Cyber Security Resilience (CSR)*, 2021, pp. 485–490.

[108] H. Rasheed, A. Hadi, and M. Khader, "Threat hunting using GRR rapid response," in *Proc. IEEE Int. Conf. New Trends Comput. Sci. (ICTCS)*, 2017, pp. 155–160.

[109] K. Satvat, R. Gjomemo, and V. Venkatakrishnan, "EXTRACTOR: Extracting attack behavior from threat reports," in *Proc. IEEE Eur. Symp. Security Privacy (EuroS P)*, 2021, pp. 598–615.

[110] B. Bhattarai and H. Huang, "SteinerLog: Prize collecting the audit logs for threat hunting on enterprise network," in *Proc. ACM Asia Conf. Comput. Commun. Security (AsiaCCS)*, 2022, pp. 97–108.

[111] E. Pelofske, L. M. Liebrock, and V. Urias, "Cybersecurity threat hunting and vulnerability analysis using a Neo4j graph database of open source intelligence," 2023, *arXiv:2301.12013*.

[112] B. E. Ujcich, S. Jero, R. Skowyra, A. Bates, W. H. Sanders, and H. Okhravi, "Causal analysis for software-defined networking attacks," in *Proc. USENIX Security Symp. (USENIX Security)*, 2021, pp. 3183–3200.

[113] A. Tabiban, H. Zhao, Y. Jarraya, M. Pourzandi, M. Zhang, and L. Wang, "ProvTalk: Towards interpretable multi-level provenance analysis in networking functions virtualization (NFV)," in *Proc. Netw. Distrib. Syst. Security (NDSS)*, 2022, pp. 1–18.

[114] K. Pei et al., "HERCULE: Attack story reconstruction via community discovery on correlated log graph," in *Proc. Annu. Conf. Comput. Security Appl. (ACSAC)*, 2016, pp. 583–595.

[115] A. Berady, M. Jaume, V. V. T. Tong, and G. Guette, "From TTP to IoC: Advanced persistent graphs for threat hunting," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 2, pp. 1321–1333, Jun. 2021.

[116] Z. Li, J. Zeng, Y. Chen, and Z. Liang, "AttacKG: Constructing technique knowledge graph from cyber threat intelligence reports," in *Proc. ESORICS*, 2022, pp. 589–609.

[117] Z. Jadidi and Y. Lu, "A threat hunting framework for industrial control systems," *IEEE Access*, vol. 9, pp. 164118–164130, 2021.

[118] R. Puzis, P. Zilberman, and Y. Elovici, "ATHAFI: Agile threat hunting and forensic investigation," 2020, *arXiv:2003.03663*.

[119] M. Dehghan, B. Sadeghiyan, E. Khosravian, A. S. Moghaddam, and F. Nooshi, "ProAPT: Projection of APT threats with deep reinforcement learning," 2022, *arXiv:2209.07215*.

[120] S. Qamar, Z. Anwar, M. A. Rahman, E. Al-Shaer, and B.-T. Chu, "Data-driven analytics for cyber-threat intelligence and information sharing," *Comput. Security*, vol. 67, pp. 35–58, Jun. 2017.

[121] P. Gao et al., "Enabling efficient cyber threat hunting with cyber threat intelligence," in *Proc. IEEE Int. Conf. Data Eng. (ICDE)*, 2021, pp. 193–204.

[122] P. Gao et al., "A system for efficiently hunting for cyber threats in computer systems using threat intelligence," in *Proc. IEEE Int. Conf. Data Eng. (ICDE)*, 2021, pp. 2705–2708.

[123] P. Karuna, E. Hemberg, U.-M. O'Reilly, and N. Rutar, "Automating cyber threat hunting using NLP, automated query generation, and genetic perturbation," 2021, *arXiv:2104.11576*.

[124] A. Alsaheel et al., "ATLAS: A sequence-based learning approach for attack investigation," in *Proc. USENIX Security Symp. (USENIX Security)*, 2021, pp. 3005–3022.

[125] N. Afzaliseresht, Y. Miao, S. Michalska, Q. Liu, and H. Wang, "From logs to stories: Human-centred data mining for cyber threat intelligence," *IEEE Access*, vol. 8, pp. 19089–19099, 2020.

[126] M. Villarreal-Vasquez, G. M. Howard, S. Dube, and B. Bhargava, "Hunting for insider threats using LSTM-based anomaly detection," *IEEE Trans. Depend. Secure Comput.*, vol. 20, no. 1, pp. 451–462, Jan./Feb. 2023.

[127] S.-X. Lin, Z.-J. Li, T.-Y. Chen, and D.-J. Wu, "Attack tactic Labeling for cyber threat hunting," in *Proc. Int. Conf. Adv. Commun. Technol. (ICACT)*, 2022, pp. 34–39.

[128] C.-K. Chen, S.-C. Lin, S.-C. Huang, Y.-T. Chu, C.-L. Lei, and C.-Y. Huang, "Building machine learning-based threat hunting system from scratch," *Digit. Threats Res. Pract.*, vol. 3, no. 3, pp. 1–21, 2022.

[129] S. Schmitt, F. I. Kandah, and D. Brownell, "Intelligent threat hunting in software-defined networking," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, 2019, pp. 1–5.

[130] M. Abdel-Basset, H. Hawash, and K. Sallam, "Federated threat-hunting approach for microservice-based industrial cyber-physical system," *IEEE Trans. Ind. Informat.*, vol. 18, no. 3, pp. 1905–1917, Mar. 2022.
[131] K. Dempsey, P. Eavy, and G. Moore, *Automation Support for Security Control Assessments*, Nat. Inst. Stand. Technol., Gaithersburg, MD, USA, 2017.
[132] M. Iannacone et al., "Developing an ontology for cyber security knowledge graphs," in *Proc. Annu. Cyber Inf. Security Res. Conf.*, 2015, pp. 1–4.
[133] S. Barnum, *Standardizing Cyber Threat Intelligence Information With the Structured Threat Information Expression (STIX)*, Mitre Corporat., Bedford, MA, USA, 2012.
[134] T. C. Truong, I. Zelinka, J. Plucar, M. Čandík, and V. Šulc, "Artificial intelligence and cybersecurity: Past, presence, and future," in *Artificial Intelligence and Evolutionary Computations in Engineering Systems*. Singapore: Springer, 2020, pp. 351–363.
[135] H. Binyamini, R. Bitton, M. Inokuchi, T. Yagyu, Y. Elovici, and A. Shabtai, "An automated, end-to-end framework for modeling attacks from vulnerability descriptions," 2020, *arXiv:2008.04377*.
[136] A. Boukhtouta, T. Madi, M. Pourzandi, and H. Alameddine, "Cloud native applications profiling using a graph neural networks approach," in *Proc. IEEE Future Netw. World Forum (FNWF)*, 2022, pp. 220–227.
[137] D. B. Rawat, R. Doku, and M. Garuba, "Cybersecurity in big data era: From securing big data to data-driven security," *IEEE Trans. Services Comput.*, vol 14, no. 6, pp. 2055–2072, Dec. 2021.
[138] V. S. Sree, C. S. Koganti, S. K. Kalyana, and P. Anudeep, "Artificial intelligence based predictive threat hunting in the field of cyber security," in *Proc. Global Conf. Advancement Technol. (GCAT)*, 2021, pp. 1–6.
[139] A. Yazdinejad, M. Kazemi, R. M. Parizi, A. Dehghantanha, and H. Karimipour, "An ensemble deep learning model for cyber threat hunting in Industrial Internet of Things," *Digit. Commun. Netw.*, vol. 9, no. 1, pp. 101–110, 2023.
[140] M. Beechey, K. G. Kyriakopoulos, and S. Lambotharan, "Evidential classification and feature selection for cyber-threat hunting," *Knowl. Based Syst.*, vol. 226, Aug. 2021, Art. no. 107120.
[141] F. O. Olowononi, D. B. Rawat, and C. Liu, "Resilient machine learning for networked cyber physical systems: A survey for machine learning security to securing machine learning for CPS," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 1, pp. 524–552, 1st Quart., 2020.
[142] *Cortex XDR*, Paloalto Netw., Santa Clara, CA, USA, 2022.
[143] *Introducing IBM Security QRadar XDR*, IBM, Armonk, NY, USA, 2022.
[144] *Adopting Splunk's Analytics-Driven Security Platform as Your SIEM*, Splunk, San Francisco, CA, USA, 2020.
[145] X.-H. Li et al., "A survey of data-driven and knowledge-aware explainable AI," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 1, pp. 29–49, Jan. 2022.
[146] S. Neupane et al., "Explainable intrusion detection systems (X-IDS): A survey of current methods, challenges, and opportunities," *IEEE Access*, vol. 10, pp. 112392–112415, 2022.
[147] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: A survey," *Data Min. Knowl. Disc.*, vol. 29, no. 3, pp. 626–688, 2015.
[148] S. Lee, H. Cho, N. Kim, B. Kim, and J. Park, "Managing cyber threat intelligence in a graph database: Methods of analyzing intrusion sets, threat actors, and campaigns," in *Proc. Int. Conf. Platform Technol. Service (PlatCon)*, 2018, pp. 1–6.
[149] Y. Jia, Y. Qi, H. Shang, R. Jiang, and A. Li, "A practical approach to constructing a knowledge graph for cybersecurity," *Engineering*, vol. 4, no. 1, pp. 53–60, 2018.
[150] K. Liu, F. Wang, Z. Ding, S. Liang, Z. Yu, and Y. Zhou, "A review of knowledge graph application scenarios in cyber security," 2022, *arXiv:2204.04769*.
[151] D. S. Johnson, M. Minkoff, and S. Phillips, "The prize collecting Steiner tree problem: Theory and practice," in *Proc. SODA*, vol. 1, 2000, p. 4.
[152] *XDR at a Glance*, Cisco Secure, San Jose, CA, USA, 2022.
[153] P. Parameshwarappa, Z. Chen, and A. Gangopadhyay, "Analyzing attack strategies against rule-based intrusion detection systems," in *Proc. Workshop Program Int. Conf. Distrib. Comput. Netw. (ICDCN)*, 2018, pp. 1–4.
[154] D. Hermawan, N. G. Novianto, and D. Octavianto, "Development of open source-based threat hunting platform," in *Proc. IEEE Int. Conf. Artif. Intell. Data Sci. (AiDAS)*, 2021, pp. 1–6.
[155] J. Chen and M. Talha, "Audit data analysis and application based on correlation analysis algorithm," *Comput. Math. Methods Med.*, vol. 2021, Nov. 2021, Art. no. 2059432.
[156] G. Mühl, L. Fiege, and P. Pietzuch, *Distributed Event-Based Systems*. New York, NY, USA: Springer, 2006.
[157] V. Mavroeidis and S. Bromander, "Cyber threat intelligence model: An evaluation of taxonomies, sharing standards, and ontologies within cyber threat intelligence," in *Proc. Eur. Intell. Security Inf. Conf. (EISIC)*, 2017, pp. 91–98.
[158] *SIEM Capabilities Through FireEye Helix*, FireEye, Milpitas, CA, USA, 2019.
[159] M. Iaiani, A. Tugnoli, S. Bonvicini, and V. Cozzani, "Analysis of cybersecurity-related incidents in the process industry," *Rel. Eng. Syst. Safety*, vol. 209, May 2021, Art. no. 107485.
[160] N. Hurley, Z. Cheng, and M. Zhang, "Statistical attack detection," in *Proc. 3rd ACM Conf. Recommender Syst.*, 2009, pp. 149–156.
[161] A. Georgiadou, S. Mouzakitis, and D. Askounis, "Assessing MITRE ATT&CK risk using a cyber-security culture framework," *Sensors*, vol. 21, no. 9, p. 3267, 2021.
[162] R. Yamagishi, T. Katayama, N. Kawaguchi, and T. Shigemoto, "HOUND: Log analysis support for threat hunting by log visualization," in *Proc. IEEE Int. Congr. Adv. Appl. Inf. (IIAI-AAI)*, 2022, pp. 653–656.
[163] M. Al Shibani and E. Anupriya, "Automated threat hunting using ELK stack—A case study," *Indian J. Comput. Sci. Eng*, vol. 10, no. 5, pp. 118–127, 2019.
[164] K. Subramanian and W. Meng, "Threat hunting using elastic stack: An evaluation," in *Proc. IEEE Int. Conf. Service Oper. Logist. Inf. (SOLI)*, 2021, pp. 1–6.
[165] *ArcSight's Latest and Greatest*, Micro Focus, Newbury, U.K., 2022.
[166] *CrowdStrike Falcon XDR*, CrowdStrike, Sunnyvale, CA, USA, 2022.
[167] A. Ramsdale, S. Shiaeles, and N. Kolokotronis, "A comparative analysis of cyber-threat intelligence sources, formats and languages," *Electronics*, vol. 9, no. 5, p. 824, 2020.
[168] *Information Technology—Security Techniques—Selection, Deployment and Operations of Intrusion Detection and Prevention Systems (IDPS)*, ISO/IEC Standard 27039, 2015.
[169] *Information Security, Cybersecurity and Privacy Protection—Information Security Management Systems—Requirements*, ISO/IEC Standard 27001, 2022.
[170] *Information Security, Cybersecurity and Privacy Protection—Information Security Controls*, ISO/IEC Standard 27002, 2022.
[171] *Information Technology—Information Security Incident Management*, ISO/IEC Standard 27035, 2023.
[172] *Information Security, Cybersecurity and Privacy Protection—Guidance on Managing Information Security Risks*, ISO/IEC Standard 27005, 2022.
[173] P. Agbabia, *Understanding the Open Cybersecurity Schema Framework*, IETF, Fremont, CA, USA, 2022. [Online]. Available: https://schema.ocsf.io/
[174] K. A. Akbar, Y. Wang, M. S. Islam, A. Singhal, L. Khan, and B. Thuraisingham, "Identifying tactics of advanced persistent threats with limited attack traces," in *Proc. Int. Conf. Inf. Syst. Security (ICISS)*, 2021, pp. 3–25.
[175] *Open Command and Control (OpenC2)*, Org. Adv. Struct. Inf., Columbus, OH, USA, 2021.
[176] *Kestrel Threat Hunting Language*, IBM, Armonk, NY, USA, 2021.
[177] *STIX-Shifter*, IBM, Armonk, NY, USA, 2021.
[178] Microsoft Threat Intelligence. "MSTIC Jupyter and python security tools." 2022. [Online]. Available: https://github.com/microsoft/msticpy
[179] J. Connolly, M. Davidson, and C. Schmidt, *The Trusted Automated Exchange of Indicator Information (TAXII)*, MITRE, McLean, VA, USA, 2014.
[180] C. Wagner, A. Dulaunoy, G. Wagener, and A. Iklody, "MISP: The design and implementation of a collaborative threat intelligence sharing platform," in *Proc. ACM Workshop Inf. Sharing Collaborative Security*, 2016, pp. 49–56.
[181] A. Dulaunoy and A. Iklody, "MISP core format," Internet Eng. Task Force, Fremont, CA, USA, Feb. 2022.
[182] O. A. Waraga, M. Bettayeb, Q. Nasir, and M. A. Talib, "Design and implementation of automated IoT security testbed," *Comput. Security*, vol. 88, Jan. 2020, Art. no. 101648.
[183] M. Arafune et al., "Design and development of automated threat hunting in industrial control systems," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, 2022, pp. 618–623.

[184] J. Elgh, "Comparison of adversary emulation tools for reproducing behavior in cyber attacks," M.S. thesis, Dept. Comput. Inf. Sci., Linköping Univ., Linköping, Sweden, 2022.
[185] S. Siboni et al., "Security testbed for Internet-of-Things devices," *IEEE Trans. Rel.*, vol. 68, no. 1, pp. 23–44, Mar. 2019.
[186] S. Choi, J. Choi, J.-H. Yun, B.-G. Min, and H. Kim, "Expansion of ICS testbed for security validation based on MITRE attack techniques," in *Proc. 13th USENIX Conf. Cyber Security Exp. Test*, 2020, p. 2.
[187] C. Alcaraz and J. Lopez, "Digital twin: A comprehensive survey of security threats," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 3, pp. 1475–1503, 3rd Quart., 2022.
[188] R. Faleiro, L. Pan, S. R. Pokhrel, and R. Doss, "Digital twin for cybersecurity: Towards enhancing cyber resilience," in *Proc. EAI Int. Conf. Broadband Commun. Netw. Syst. (BROADNETS)*, 2022, pp. 57–76.
[189] A. Pokhrel, V. Katta, and R. Colomo-Palacios, "Digital twin for cybersecurity incident prediction: A multivocal literature review," in *Proc. IEEE/ACM 42nd Int. Conf. Softw. Eng. Workshops*, 2020, pp. 671–678.
[190] S. Pirbhulal, H. Abie, and A. Shukla, "Towards a novel framework for reinforcing cybersecurity using digital twins in IoT-based healthcare applications," in *Proc. IEEE 95th Veh. Technol. Conf. (VTC-Spring)*, 2022, pp. 1–5.
[191] R. Candell, K. Stouffer, and D. Anand, "A cybersecurity testbed for industrial control systems," in *Proc. Process Control Safety Symp.*, 2014, pp. 1–16.
[192] B. Green et al., "ICS testbed tetris: Practical building blocks towards a cyber security resource," in *Proc. 13th USENIX Workshop Cyber Security Exp. Test (CSET)*, 2020, pp. 1–9.
[193] O. Wenge, U. Lampe, C. Rensing, and R. Steinmetz, "Security information and event monitoring as a service: A survey on current concerns and solutions," *PIK-Praxis der Informationsverarbeitung und Kommunikation*, vol. 37, no. 2, pp. 163–170, 2014.
[194] *Cortex XSOAR: Redefining Security Orchestration, Automation, and Response*, Paloalto Netw., Santa Clara, CA, USA, 2023.
[195] P. Firstbrook and C. Lawson, "Innovation insight for extended detection and response," Gartner, Stamford, CO, USA, Rep. G00718616, 2021.
[196] *Falcon Insight: Endpoint Detection and Response (EDR)*, CrowdStrike, Sunnyvale, CA, USA, 2022.
[197] *Nowhere to Hide: 2021 Threat Hunting Report*, CrowdStrike, Sunnyvale, CA, USA, 2022.
[198] *X-Force Threat Intelligence Index 2022*, IBM, Armonk, NY, USA, 2022.
[199] H. Kaur and H. S. Pannu, "Anomaly detection survey for information security," in *Proc. Int. Conf. Security Inf. Netw. (SIN)*, 2017, pp. 251–258.
[200] S. Yuan and X. Wu, "Deep learning for insider threat detection: Review, challenges and opportunities," *Comput. Security*, vol. 104, May 2021, Art. no. 102221.
[201] Y. Deng, D. Lu, D. Huang, C.-J. Chung, and F. Lin, "Knowledge graph based learning guidance for cybersecurity hands-on labs," in *Proc. ACM Conf. Global Comput. Educ.*, 2019, pp. 194–200.
[202] O. Gheibi, D. Weyns, and F. Quin, "Applying machine learning in self-adaptive systems: A systematic literature review," *ACM Trans. Auton. Adapt. Syst.*, vol. 15, no. 3, pp. 1–37, 2021.
[203] R. U. Islam, M. S. Hossain, and K. Andersson, "A deep learning inspired belief rule-based expert system," *IEEE Access*, vol. 8, pp. 190637–190651, 2020.
[204] E. Bout, V. Loscri, and A. Gallais, "How machine learning changes the nature of cyberattacks on IoT networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 248–279, 1st Quart., 2021.
[205] I. H. Sarker, A. Kayes, S. Badsha, H. Alqahtani, P. Watters, and A. Ng, "Cybersecurity data science: An overview from machine learning perspective," *J. Big Data*, vol. 7, no. 1, pp. 1–29, 2020.
[206] P. Rajesh, M. Alam, M. Tahernezhadi, A. Monika, and G. Chanakya, "Analysis of cyber threat detection and emulation using MITRE attack framework," in *Proc. Int. Conf. Intell. Data Sci. Technol. Appl. (IDSTA)*, 2022, pp. 4–12.
[207] J. A. Kroll, J. B. Michael, and D. B. Thaw, "Enhancing cybersecurity via artificial intelligence: Risks, rewards, and frameworks," *Computer*, vol. 54, no. 6, pp. 64–71, 2021.
[208] X. Chen, S. Jia, and Y. Xiang, "A review: Knowledge reasoning over knowledge graph," *Exp. Syst. Appl.*, vol. 141, Mar. 2020, Art. no. 112948.
[209] S. Yoo and O. Jeong, "Automating the expansion of a knowledge graph," *Exp. Syst. Appl.*, vol. 141, Mar. 2020, Art. no. 112965.
[210] S. E. Ooi et al., "Intent-driven secure system design: Methodology and implementation," *Comput. Security*, vol. 124, Jan. 2023, Art. no. 102955.
[211] J. Kim et al., "IBCS: Intent-based cloud services for security applications," *IEEE Commun. Mag.*, vol. 58, no. 4, pp. 45–51, 2020.
[212] N. DeCastro-García and E. Pinto, "Measuring the quality information of sources of cybersecurity by multi-criteria decision making techniques," in *Proc. Int. Conf. Hybrid Artif. Intell. Syst.*, 2022, pp. 75–87.
[213] S. Myneni et al., "DAPT 2020-constructing a benchmark dataset for advanced persistent threats," in *Proc. 1st Int. Workshop Deployable Mach. Learn. Security Defense (MLHat)*, Aug. 2020, pp. 138–163.
[214] G. Sebastian, "Do ChatGPT and other AI chatbots pose a cybersecurity risk? An exploratory study," *Int. J. Security Privacy Pervasive Comput.*, vol. 15, no. 1, pp. 1–11, 2023.
[215] D. M. Manias and A. Shami, "The need for advanced intelligence in NFV management and orchestration," *IEEE Netw.*, vol. 35, no. 1, pp. 365–371, Jan./Feb. 2021.

Boubakr Nour received the Ph.D. degree in computer science and technology from the Beijing Institute of Technology, China. He is a Researcher with Ericsson, Canada. His research interests include threat hunting, proactive security, next-generation networking, and Internet.

Makan Pourzandi received the M.Sc. degree in computer science from the Ecole Normale Superieure de Lyon, France, and the Ph.D. degree in computer science from the University of Lyon, France. He is a Researcher with Ericsson, Canada. His current research interests include security, cloud computing, software security engineering, cluster computing, and component-based methods for secure software development.

Mourad Debbabi received the B.Eng. degree from the Université de Constantine, and the M.Sc. and Ph.D. degrees in computer science from Paris-XI Orsay University, France. He is a Full Professor with the Concordia Institute for Information Systems Engineering and the Dean of the Gina Cody School of Engineering and Computer Science, Concordia University. He holds the NSERC/Hydro-Quebec Thales Senior Industrial Research Chair in Smart Grid Security and the Hon. Concordia Research Chair Tier I in Information Systems Security. He is a Founding Member and the Executive Director of the National Cybersecurity Consortium that leads the CyberSecurity Innovation Network Program. He serves on the expert committee of the Ministry of Cybersecurity and Digital Technology of the Quebec Government. He serves/served on the boards of the Canadian Police College, PROMPT Québec, Cybereco, and Calcul Québec. He served as a member of CATAAlliance's Cybercrime Advisory Council. He is the Founder and the Director of the Security Research Centre, Concordia University. He served as a Senior Scientist with the Panasonic Information and Network Technologies Laboratory, Princeton, NJ, USA; an Associate Professor with the Computer Science Department, Laval University, Canada; a Senior Scientist with the General Electric Research Center, New York, NY, USA; a Research Associate with the Computer Science Department, Stanford University, CA, USA; and a Permanent Researcher with the Bull Corporate Research Center, Paris, France. He supervised to successful completion 34 Ph.D. students, 76 master's students, and 15 Postdoctoral Fellows. He published seven books and more than 300 peer-reviewed research articles in international journals and conferences on cybersecurity, cyber forensics, smart grid security, privacy, cryptographic protocols, cyber threat intelligence, malware analysis, reverse engineering, specification, and verification of safety-critical systems, programming languages, and type theory.