



APT beaconing detection: A systematic review

Manar Abu Talib^a, Qassim Nasir^b, Ali Bou Nassif^b, Takua Mokhamed^a, Nafisa Ahmed^{a,*}, Bayan Mahfood^a

^a Department of Computer Science, College of Computing and Informatics, University of Sharjah, United Arab Emirates

^b Department of Electrical and Computer Engineering, College of Computing and Informatics, University of Sharjah, United Arab Emirates

ARTICLE INFO

Article history:

Received 29 July 2021

Revised 7 July 2022

Accepted 12 August 2022

Available online 21 August 2022

Keywords:

APT
Beaconing
Attack
Security breach
Detection
AI
C&C

ABSTRACT

Advanced Persistent Threat (APT) is a type of threat that has grabbed the attention of researchers, particularly in the industrial security field. APTs are cyber intrusions carried out by skilled and well-resourced adversaries who target specific information in high-profile organizations and governments, frequently as part of a multi-phase long-term operation. One of the phases of the APT process is the command-and-control (C&C) phase, also known as beaconing. Beaconing is an important part of an APT lifecycle, where the adversaries establish channels with the compromised hosts in the targeted system, allowing them to launch additional attacks. Detecting and predicting this stage is therefore a practical way to guard against APTs. This paper discusses the techniques and methods used to detect APTs and also specifically to identify beaconing, either during the APT lifecycle or not. In it, we determine various artificial intelligence algorithms used for detecting, analyzing and comparing characteristics of datasets and data sources used to implement these detection techniques. Moreover, we present the strengths and challenges of various APT/beaconing detection methods. Finally, this study outlines many cybersecurity vendor projects that have been created to identify APT or beaconing operations, categorized according to the detection approach utilized.

© 2022 Elsevier Ltd. All rights reserved.

1. Introduction

The rise in power and popularity of the internet has increased the number and influence of cyber attackers. Many corporations and companies have tried to keep malware and unwelcome invaders away for years with varying levels of success (Li et al., 2016). As a result, cyber-attackers have invented increasingly sophisticated ways to circumvent security systems. APTs are an advanced variant of these cyberattacks; they require complex tools as well as specialists with a high degree of expertise to execute them. They are sophisticated in nature, long-term and persistent (Li et al., 2016). The term "Advanced Persistent Threat" accurately describes the main characteristics of this type of attack (Vukalović and Delija, 2015):

APTs are advanced attacks, which means that they are covert, targeted, and data-focused, with attackers continually adjusting their approaches if they fail to achieve their goal, which is generally the extraction of sensitive or important data. Additionally, APT

attacks generally have excellent stealth capabilities. The attackers' entry, tactics, and timing are all unexpected and imprecise, making it challenging for standard detection methods to identify them.

APTs are persistent in nature, meaning that the attackers maintain a long-term network presence rather than causing immediate system harm. The longest analyzed assault by Chinese espionage team the APT1 group lasted around four years and ten months, according to McWhorte (2013).

APTs are a threat: they aim to extract sensitive data such as strategic intelligence about a corporation or a business. As a result, APT assaults frequently cause significant harm to the target (Stojanović et al., 2020).

Considering these main APT characteristics, we can conclude that the whole purpose of an APT attack is to gain ongoing access to a target system. The attackers accomplish this goal in a series of stages, which are considered parts of the APT lifecycle. There are several proposed versions of the APT lifecycle (Li et al., 2016; Vukalović and Delija, 2015; Brewer, 2014; Ussath et al., 2016; Messaoud et al., 2016; Virvilis et al., 2013), yet they share the same common APT steps. We consider the six stages proposed by Chen et al. (2014) (Fig. 1) to be particularly representative of the phenomenon. It is explained below:

Reconnaissance and Weaponization: Gathering information about the target organization. APT actors create an attack plan

* Corresponding author.

E-mail addresses: mtalib@sharjah.ac.ae (M. Abu Talib), nasir@sharjah.ac.ae (Q. Nasir), anassif@sharjah.ac.ae (A. Bou Nassif), tmokhamed@sharjah.ac.ae (T. Mokhamed), nafisa.ahmed@sharjah.ac.ae, nafisa.elrasheed@gmail.com (N. Ahmed), bmahfood@sharjah.ac.ae (B. Mahfood).

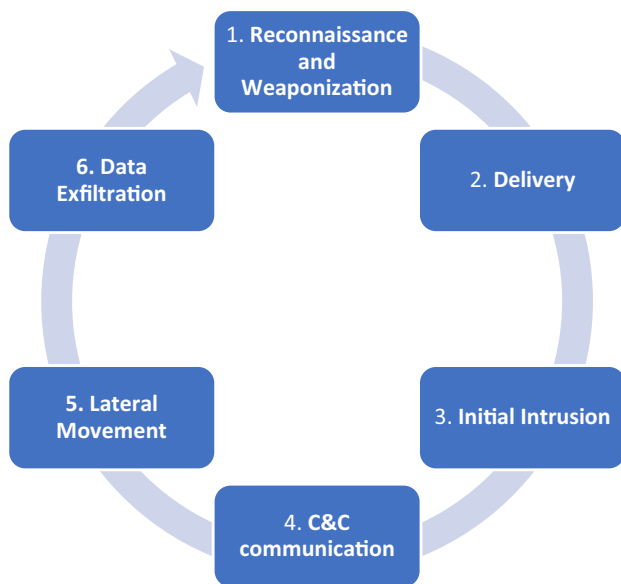


Fig. 1. APT lifecycle.

and prepare the appropriate equipment based on the knowledge acquired. Attackers often prepare equipment for several different styles of attack, allowing them to adjust their strategies in the event of failure while increasing their rate of success.

Delivery: The attackers send exploits to the targets, using direct or indirect delivery methods. The attackers use social engineering tactics like spear-phishing to convey exploits to their targets for direct delivery. Indirect delivery is unobtrusive. The attackers will compromise a trusted third party and then utilize it to exploit the victim indirectly.

Initial Intrusion: The attackers exploit an entry point, gain a foothold and establish an outbound connection. APT attackers employ a variety of tools and strategies to exploit vulnerabilities discovered in the target organization's online applications, while also exploiting vulnerabilities in end-user computers via malware execution. After accessing the targeted network, APT attackers seek to develop a command-and-control (C&C) communication channel through which they can launch additional attacks.

C&C communication: In this stage, attackers consolidate their presence at the entry points and take control of compromised computers, enabling further exploitation of the network. Attackers utilize a variety of tactics to gain access to critical resources and other hosts within the compromised system.

Lateral Movement: In this stage, the attackers compromise other hosts in the network to discover and gather valuable data.

Data Exfiltration: In this crucial stage, the attackers export the data they collected from the internal network to their command-and-control server.

In each of these stages, the attack can be recognized in different ways and with various probabilities. APT attack detection is therefore a very challenging task. As stated earlier, APTs go through a command-and-control (C&C) communication phase, also known as C&C, C2, or beaconing. This is an important part of an APT's lifecycle in which the adversaries establish channels with compromised hosts in the targeted system, allowing them to launch additional attacks (Alshamrani et al., 2019). The most crucial part of this attack stage involves the invaders setting up open communication, allowing them to access sensitive resources and obtain the information they seek (Ussath et al., 2016). An accurate and efficient detection method is required to increase the level of security of the target organization, thus protecting its data from vulnerability.

According to recent systematic surveys, various detection methodologies and strategies have been developed to protect against APT attacks based on deep or machine learning methods (S. Quintero-Bonilla and del Rey, 2020; Rajalakshmi et al., 2019; S. Quintero-Bonilla and del Rey, 2020) and behavior pattern analysis (Singh et al., 2019). However, many of these systematic surveys detect only one stage of APT attacks. In addition, they do not focus on beaconing activities during APTs, which give the attackers open access to the organization's resources and create fundamental security problems. As a result, there is an urgent need to conduct a detailed study on possible APT beaconing detection solutions that can guarantee the safety of target organizations. This is the issue at the heart of this work.

The main goal of this study is to provide a systematic review and to perform detailed research into various APT-specific detection techniques and solutions. Furthermore, the techniques and strategies that focus on detecting command-and-control (C&C or C2) malware, and beaconing during a targeted APT are closely examined. We highlight the Artificial Intelligence algorithms and dataset characteristics widely used to detect beaconing behavior. We also analyze the strengths and weaknesses of each beaconing detection technique. We believe that this review will help researchers better understand different APT beaconing detection strategies and place them in context with recent research on the topic. We also discuss APT detection vendor projects that are currently using these strategies.

The rest of the paper is organized as follows: Section 2 discusses the literature review. Section 3 illustrates the methodology used to conduct the Systematic Literature Review (SLR), which consists of planning and conducting the research. In Section 4, we discuss the results of our findings on the predefined research questions. Finally, Section 5 presents the conclusion and possibilities for future work.

2. Literature review

Adopting a specific mechanism to defend against APTs is difficult, due to the rapid evolution of threat tools and techniques. Attackers are always looking for new ways to get into their targets' systems. Each day brings new types of malware, along with new signatures, activities, and behaviors similar to normal, so that a single threat identification mechanism is not sufficient (Vukalović and Delija, 2015). As a result, one of the most challenging aspects of threat detection technology is recognizing, predicting, and identifying the various types of APT attacks with their continuously changing behavior (Vukalović and Delija, 2015). Motivated by the difficulty of this problem, we searched for articles and papers that addressed the various approaches used in detecting APTs. For example, Alshamrani et al. (2019) provided a comprehensive study of the APT lifecycle. They reviewed all known APT detection tools used to identify various stages of APT attacks and studied learning approaches that could be used to make the threat detection system smart enough to identify adapting APT attacks. Furthermore, this paper discusses various challenges to defending against APTs. In another example, Quintero-Bonilla and del Rey (2020a) presented an extensive survey of APT detection techniques focusing on machine learning mechanisms and the lifecycle of the attack. They introduced the area of APT attack detection research, delved into the background of the problem, and defined and discussed various techniques and algorithms. Moreover, they presented existing machine learning solutions for the detection of APTs in two main categories: supervised and unsupervised approaches. Similarly, Rajalakshmi et al. (2019) studied numerous Machine Learning algorithms and techniques for detecting Advanced Persistent Threats, while Stojanović et al. (2020) reviewed the literature on datasets and their construction for use in

APT detection, paying special attention to feature engineering, involving construction, selection and dimension reduction. As these datasets are built on an attack model, an overview of the various phases of such attacks, including methods and targets, is given. In addition, the definition and comparison of current feature extraction methodologies, as well as a comprehensive review of datasets used in APT detection-related literature, are presented in this paper. Lemay et al. (2018) proposed a survey of open-source literature on APT actors and their activities, emphasizing APT activities rather than studies on defense or detection approaches. The writer aims to provide a simple guide to the state of APT actors' expertise, allowing interested researchers to determine which primary sources are most important to their study. The paper includes publications from about 40 APT communities from various parts of the world. Each publication's key findings are summarized in a brief overview. Moreover, Quintero-Bonilla and del Rey (2020b) surveyed the machine learning techniques and algorithms used in different frameworks or models that detect and predict APT attacks. The paper also provides a brief analysis of the components and design of the framework and APT lifecycle of proposed models. The authors found that the machine learning algorithms used in the proposed models are supervised learning algorithms. In another paper, Singh et al. (2019) introduced a systematic analysis of semantic-aware work to identify possible contributions exploring and detecting APT in greater depth. Further, the authors describe the modeling phase and behavioral pattern that characterizes the usual steps taken by APT attackers to gather the requested information. In addition, the paper contains some recent zero-day threats, use cases, and cyber developments in the South-east. The study introduces a rigorous literature assessment system that classifies APT attack activities and suggests preventive measures. The research further discusses potential study directions for APT security systems in the context of the next-generation threat lifecycle. Nissim et al. (2015) conducted a survey identifying methods, procedures, and tools used to detect suspicious PDF files. According to this study, these PDFs are often attached to e-mails sent to organizations to carry out the initial penetration of an APT attack; their identification is a major problem that needs attention. Luh et al. (2017) provide a detailed overview of possible techniques, strategies, models, structures, methodologies, and systems that could be useful in defending against APTs and other multi-stage cyber-attacks. This paper gives a comprehensive analysis of the four levels of situational and organizational threat intelligence, and the various solutions currently under investigation: general or supporting solutions, host-based solutions, network-based solutions and multi-source solutions. Finally, Ahmad et al. (2019) proposed a study that examines the use of the term 'APT' as well as the origin and evolution of the concept, and determines the term's formal definition. Strategically motivated APTs, or S-APTs, are a type of APT described by the authors. S-APTs differ from other APTs in that they derive their goals from third-party strategic agendas, according to a basic typology. The APT Operation Line (APTOL) model is then used to present an operating architecture for understanding advanced persistent threats (Ahmad et al., 2019). In addition, the authors describe how S-APTs use TTP to carry out their strategic operations. The role of human situation awareness in these operations is examined, along with how it can be used as a weapon for counterattack.

In this Systematic Literature Review, we present findings that are distinct from those in the surveys listed above. Our research is unique in that it provides a comprehensive analysis of all proposed APT beaconing detection approaches and solutions, as well as a precise comparison of each solution's strengths and weaknesses. Finally, we go through the Artificial Intelligence algorithms and datasets used to implement solutions for detecting APT/beaconing attacks.

3. Methodology

Kitchenham and Charters' methods (Keele, 2007) directed us in conducting a Systematic Literature Review (SLR). Planning, execution, and reporting are the three major phases of this method. Multiple processes and steps are included in each stage. The following six phases are used in the planning stage: determine your research questions, identify your search strategy, develop criteria for your selection, establish your quality assessment rules, define the data extraction methods, and define how the extracted data can be synthesized. A detailed overview of the steps will be provided in the following subsection.

3.1. Research questions

The primary aim of this study is to review the APT/beaconing detection research area. The following research questions are raised to accomplish this aim.

RQ1: What techniques are used to detect an APT attack? What Artificial Intelligence-aware algorithms are used to detect APT/beaconing behavior?

RQ1 aims to identify the strategies and solutions applied by researchers to detect APT attacks, as well as the AI techniques that have been used to recognize APT/beaconing attacks.

RQ2: What techniques can be used to detect beaconing? Which APT detection techniques focus on detecting beaconing during APT?

RQ2 is concerned with strategies used to detect beaconing attacks in general, whether or not it is part of an APT attack. It focuses on the strategies used specifically for the detection of the beaconing phase of an APT.

RQ3: What are the main characteristics of the datasets/data sources most commonly used in APT/ beaconing detection research?

RQ3 aims to recognize the characteristics and features of datasets and data sources that have been used in APT/beaconing detection research.

RQ4: What are the strengths and weaknesses of each APT/beaconing detection technique?

RQ4 aims to present the advantages and opportunities of the techniques proposed by researchers. It also aims to present challenges and difficulties faced during the detection of APT/beaconing attacks.

3.2. Search strategy

The search strategy is divided into three parts: search terms, literature resources, and the search process, discussed in detail below.

The following was our procedure for selecting our search terms: Firstly, the main search terms were defined by the research questions. Secondly, synonyms and alternate spellings were identified for major terms. The results of the search are constrained by the Boolean operators (AND and OR). The search words used in this study refer to APT/beaconing detection.

The following is a list of all the search terms that resulted.

"Advanced persistent threat" OR "Advanced persistent threats" OR "APT" AND "Detection" AND "Beaconing"

"Advanced persistent threat" OR "Advanced persistent threats" OR "APT" AND "Detection" AND "Command and Control"

"Advanced persistent threat" OR "Advanced persistent threats" OR "APT"

"Beaconing" OR "C&C" OR "C2" OR "Command and Control" AND "Detection"

To find relevant articles (published in journals and conference papers), the following digital libraries were researched: Google

Scholar, Institute of Electrical and Electronics Engineers IEEE Explorer, Association for Computing Machinery ACM Digital Library, Science Direct, Springer, Elsevier, Hindawi, Public Library of Science and the MDPI Multidisciplinary Digital Publishing Institute. Moreover, we found several journals, such as the Innovative Information Science & Technology Research Group (ISYOU) Journal, the Journal of Universal Computer Science (Technical University in Graz), and the Journal of Penerbit Akademia Baru, that met our selection criteria.

The primary relevant papers were extracted from these digital libraries using the stated search terms. A simple search of each paper's cited references also contributed to the resources available to address the research questions. In [Section 3.3](#), the inclusion criteria are described in detail. The Google document platform was used to manage the search results and papers among authors. Based on the inclusion/exclusion criteria outlined in [Section 3.3](#), we gathered 122 publications: 95 papers related to detecting APT attacks and 27 publications related to beaconing. These resources included 45 journal papers, 6 articles, 1 chapter, and 70 conference papers. In addition, we studied 31 vendor projects that may detect or identify APT.

3.3. Study selection

Based on our search terms, we found 160 science papers using our first search. The authors carried out the filtration process separately, and the findings were discussed in planned meetings to ensure that only articles relevant to our topic were included. The following are the steps in the selection and filtration process: Removing review and survey papers from the collection, removing duplicated papers in the collection, applying inclusion and exclusion criteria to candidate papers to avoid irrelevant articles, and using quality assessment rules to determine the quality of the articles, thereby ensuring the best possible answers to the research questions.

We defined a set of the inclusion and exclusion criteria used in this research study to make sure that only relevant papers are included in this review. We only included journals and conference papers that focused on detection strategies for identifying APT/beaconing, as well as studies that discuss APT/beaconing detection solutions using AI algorithms. Furthermore, we excluded papers with no clear publication information and articles that did not mention APT/beaconing detection techniques. We also excluded those that discuss malicious detection techniques that were unrelated to APT/beaconing, as well as papers that were not peer-reviewed articles.

3.4. Quality assessment rules (QARs)

The final list of papers was chosen in this phase, and the 10 QARs were used to determine article suitability in relation to the research questions. After the QARs were identified, each paper was given a score out of a total of nine. Each QAR was given a score of 1 for "fully answered", 0.75 for "over average," 0.5 for "average", 0.25 for "below average", and 0 for "not answered". The sum of the marks assigned for the 10 QARs was used to determine the paper's overall score. Only papers that scored 5 or higher on this suitability assessment were included in our study. **Error! Reference source not found.** We selected this score of 5 because it reflects the mid-point of high-quality publications and satisfies our study goals. Appendix A shows the quality scores of the articles considered.

QAR1: Are the security research goals and objectives well-defined?

QAR2: Is the APT/beaconing background clearly addressed?

QAR3: Are the APT/beaconing detection techniques used clearly defined?

QAR4: Are the methods well designed and justifiable?

QAR5: Are the strengths of the proposed methods illustrated?

QAR6: Are the limitations of the proposed methods illustrated?

QAR7: Is the evaluation of the proposed techniques discussed?

QAR8: Is the proposed technique's evaluation compared to other techniques?

QAR9: Are the data/dataset explored and identified?

QAR10: Overall, does the study enrich the academic community or industry?

3.5. Data extraction strategy and synthesis of extracted data

The final list of articles was reviewed at this stage to extract the details needed to answer the collection of research questions. We created an extraction form to extract the necessary information. Two writers were assigned the task of extraction and testing based on the extraction method. In the case of confusion or conflict between the extractor and the checker, both writers met to discuss the conflict and decide on a course of action. The extraction form consists of the following information: the title of the paper, publishers, year of publication, type of paper (whether it was from a conference or a journal) and APT detection technique used by the author. Each detection technique was then summarized for each paper, and records were kept with regard to whether the technique detects beaconing and whether it uses AI methods, along with a list of the technique's strengths and weaknesses, the datasets used and their characteristics. It's worth keeping in mind that not all of the articles gathered could contribute to all of the research questions.

We used several processes to synthesize information that would address the RQs from the data collected from the chosen papers. In addition, to handle all the research questions, we used the narrative synthesis approach. Using technologies such as pie charts, bar charts, and graphs to visualize the results is known as narrative synthesis.

4. Results and discussion

The findings of this study will be discussed in this section. It also provides an overview of the scientific papers and APT detection vendor projects chosen to address the above-mentioned research questions. The results of each research question are examined in depth in the five sections that follow. A total of 122 research papers and 31 APT and beaconing detection vendor projects were chosen. The list of these selected resources can be found in Appendix A, [Tables 8](#) and [9](#). As seen in [Fig. 2](#), the collected research articles and software vendor projects were released between 2007 and 2022. A quality evaluation rule criterion was used, as stated above, and the scores of the chosen papers are shown in [Table 10](#).

4.1. The techniques used for the detection of an APT attack and the artificial intelligence-aware algorithms used to detect APT/beaconing behavior (RQ1)

This section aims to identify the detection methods of APT attacks to address RQ1. Moreover, this section presents the Artificial Intelligence algorithms applied in the APT/beaconing detection methods and solutions proposed in these research papers. APT malware is a lengthy attack with continuously updated instructions. Various different detection methods are proposed by researchers to detect APTs in a timely fashion and minimize their damage. In this review, we defined multiple categories of APT detection methods in the selected papers.

As shown in [Table 1](#), we identified several techniques applied by researchers in the development of APT detection solutions. In this review, the most frequently used APT detection approaches

Growth of Research Papers based on years

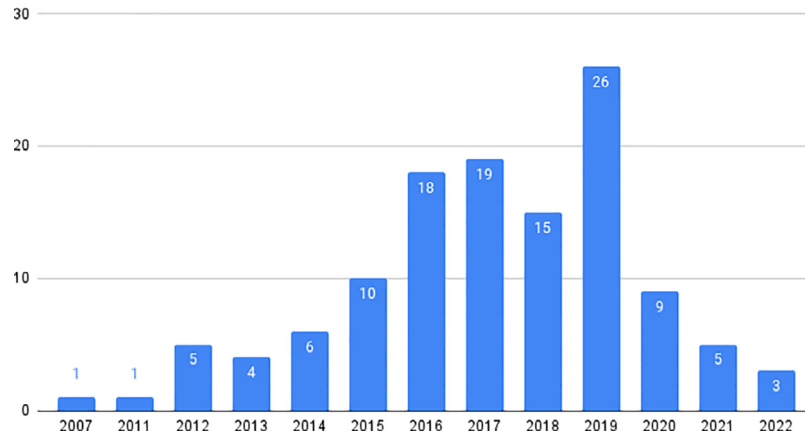


Fig. 2. Growth of scientific papers based on years.

Table 1
APT detection Techniques.

APT detection technique	Ref	Freq.	APT detection technique	Ref	Freq.
Signature-based detection	(Zhao et al., 2015; de Vries et al., 2012; Liu et al., 2012; Bencsáth et al., 2012; Liu et al., 2013; Sigholm and Bang, 2013; Najafi et al., 2021)	7	Based on the kill chain	(Bryant and Saiedian, 2017; Atapour et al., 2018; Bodström and Hämäläinen, 2018)	3
Network flow analysis-based detection	(M. Marchetti et al., 2016; M. Marchetti et al., 2016; Yan et al., 2020; Lu et al., 2019; Friedberg et al., 2015; J. Choi et al., 2015; Su et al., 2022; Wang et al., 2014; Vance, 2014; Nuojua et al., 2017; Ng and Bakhtiarib, 2016; Cho and Nam, 2019; Ghafir and Prenosil, 2016; Jia et al., 2015; Stoleriu et al., 2021)	15	Event correlation analysis	(Virvilis and Gritzalis, 2013; Ghafir et al., 2019; Giura and Wang, 2012; Mirza et al., 2014; Sharma et al., 2017; Bhatt et al., 2014; Ghafir and Prenosil, 2016; Brogi and Tong, 2016; Shan-Shan and Ya-Bin, 2018; Maccari et al., 2018)	10
Graph-based detection	(Zimba et al., 2020; Schindler, 2018; Manzoor et al., 2016; Milajerdi et al., 2019; J. Choi et al., 2015; Lamprakis et al., 2017; Rubio et al., 2017; Debatty et al., 2018; Do Xuan and Huong, 2022)	9	Honeypots	(Saudi and Islam, 2015; Lee et al., 2016)	2
Whitelisting	(Beuhring and Salous, 2014; F. Skopik et al., 2014)	2	Intrusion detection system	(Cao, 2019; Kim and Park, 2014; Friedberg et al., 2015; Cui et al., 2019)	4
Blacklisting	(Ghafir et al., 2017)	1	Disguised exe file detection (DeFD)	(I. Ghafir et al., 2018)	1
Filtering method	(Hu et al., 2016; Baksi and Upadhyaya, 2016; Kim et al., 2018; Chandra et al., 2016)	4	Based on Independent Access	(Wang et al., 2016)	1
Game-based strategy	(Huang and Zhu, 2019; Li et al., 2018; Lv et al., 2019; Zhu and Rass, 2018; Xiao et al., 2018; Hu et al., 2017; Haopu, 2016; Y. Li et al., 2019; Sengupta et al., 2019; Moothedath et al., 2020; Rass et al., 2017; Hu et al., 2015)	12	Based on memory analysis	(Ge et al., 2016; Xiong et al., 2020)	2

are the network flow analysis-based method, the signature-based detection method, the graph-based method, the game-based method and the event correlation analysis method. Other methods are less frequently used, such as blacklisting, whitelisting, and the memory analysis method. However, many of these methods can be used together with other detection methods to create powerful and efficient detection methods to recognize APT attacks.

Signature-based detection is the main method used to identify and alert on threats. This method depends on a predefined list of known indicators of APT attacks. This list of APT signatures could include the content of e-mail subject lines, malicious do-

main, malicious network attack behavior, file hashes, or known byte sequences. Signatures could also include network traffic alerts, such as known malicious IP addresses attempting to gain access to a system (Zhao et al., 2015). However, signature-based detection can only be used for known threats; it is not effective against unknown threats.

Network flow analysis-based detection, in contrast to signature-based detection, is used to identify unknown abnormal behavior. The process of anomaly-based detection involves training the detection system with normalized, standardized network behavior and then monitoring network traffic and comparing that normalized traffic to actual network activity (Lu et al., 2019).

An alarm is generated when an incident appears to be out of the norm (Cho and Nam, 2019).

The graph-based anomaly detection method is a method for finding APT anomalies in large-scale datasets. The data reveal APT attack characteristics represented as a graph. This method includes mechanisms for dealing with anomalous data that are difficult to examine using typical data mining methods (Manzoor et al., 2016). Most graph-based anomaly detection systems use a supervised approach in the selected research papers. This approach requires labeled data in advance, in order to train the system to compare normal behavior and anomalous behavior (Choi et al., 2015a). Unsupervised and semi-supervised techniques also can be used in combination with the graph-based method. Researchers proposed algorithms such as breadth-first-search (BFS), depth-first-search (DFS) and heuristic algorithms to provide a standard pattern in an input graph. Moreover, some of the research papers benefited from algorithms such as KNN and SNN to identify APT attacks (Zimba et al., 2020).

A game-based method is a technique that is based on game theory, which is a natural tool for analyzing potential conflicts of interest, such as those that occur between a defending system and an attacker launching an APT (Rass et al., 2017). This method is based on a number of distinct strategies. A generalized family of matrix games is investigated as a risk mitigation technique for advanced persistent threat (APT) defense in Rass et al. (2017). Modeling the conflict between the attacker and the defending system is a natural method. Moreover, various research papers analyzing game-based approaches, such as Sengupta et al. (2019) and Haopu (2016), used a Bayesian game strategy in which players have incomplete information about the other players. For example, a player may not know the exact payoff functions of the other players but instead have beliefs about these payoff functions (Huang and Zhu, 2019).

Honeypot is a network of honeypots with high interaction that mimics a production network and is set up in a way that allows all activity to be observed, recorded, and, to some extent, discreetly governed. The main goal is to attract APT attackers and track their movements. Honeypot servers gather information about system attackers and intrusions and then detect and analyze computer network and system intrusions.

Event Correlation Analysis is a method that examines logs or host data from across targeted networks to find correlations (relationships). Event correlation tools can then employ user-defined rules to execute actions, such as generating alerts for hardware or application problems (Virvilis and Gritzalis, 2013; Ghafir et al., 2019).

Other methods have also been adopted by some of the selected research papers, including **blacklisting and whitelisting**, **disguised exe file detection (DeFD)**, **intrusion detection-based filtering methods** and cloud computing based on **memory analysis**.

As mentioned before, many of the proposed APT and beaconing detection methods can be integrated with or fully based on other methods, especially with machine learning, deep learning, and other artificial intelligence-based methods.

Table 2 illustrates the artificial intelligence techniques used by many of the selected research papers. In addition, the table identifies which papers use each AI technique. Many research papers combined two or more deep/machine learning algorithms to improve the overall performance of the constructed detection solution. According to the table, the most frequent deep learning algorithm proposed in beaconing detection solutions is a combination of CNN and LSTM algorithms, while the deep/machine learning algorithms KNN and SVM are the most commonly used by APT detection solutions.

4.2. Beaconing and APT detection techniques used to detect beaconing during APTs (RQ2)

In this part, we will address RQ2 by introducing the various detection methods for beaconing attacks. In addition, we provide details about APT detection techniques that can identify beaconing during APT attacks.

Beaconing is the term for what happens when the infected host sends short, regular communications to an attacker to confirm that the host has been infected with malware and is ready for instructions, or ready to exfiltrate the collected data. Beacons are often delivered to command-and-control (C2 or C&C) servers outside the company network by infected internal corporate hosts. Malware administrators may use this communication approach to track, monitor, and control hundreds of thousands of infected computers automatically (Vukalović and Delija, 2015). C&C communication usually aims to imitate regular traffic patterns by using protocols such as Peer-to-Peer (P2P), Internet Relay Chat (IRC), Hyper Text Transport Protocol (HTTP), Hyper Text Transport Protocol Secure (HTTPS), Secure Shell Protocol (SSH), Domain Name System (DNS), Simple Mail Transfer Protocol (SMTP), or other customized protocols, and could also use services like Dropbox and Gmail. However, most communications between a compromised host and its C&C (Command and Control Server) use either IRC or HTTP protocols (Xing et al., 2021). It is interesting to note that each of these protocols has its own characteristics, which can serve as benefits or drawbacks for attack detectors. An attacker will identify the communication approach that is most likely to work during the information collection step, which aims to determine the sophistication of the target. They will then pre-configure their malware payloads to use the method most likely to evade typical firewall modules (Gaonkar et al., 2020). Efficient and effective detection techniques are therefore required to detect this attack. Many research papers selected for this study focus on developing a strong detection method that can accurately identify beaconing.

We collected 27 scientific papers focused on detecting C&C channels. These papers only represent beaconing; they exclude related APT papers, which are discussed later in this study. Table 3 below shows the beaconing detection methodologies proposed by these selected scientific papers, together with a description of the method and the frequency with which these methods are examined by the scientific papers. According to the table, the most frequently used techniques are behavior-based (network-based) detection, machine learning and deep learning.

Command-and-control is considered to be a critical component of the APT lifecycle. During this phase of the attack, the adversary utilizes the vulnerability of the target system. Infected systems are compelled to establish a communication link with the attacker so that they may be controlled directly. The C&C channel allows an attacker to use remote access tools to gain access to a compromised system, load further specialized malware modules, and undertake other malicious actions such as spreading to other devices (Zimba et al., 2020). In this study, we gathered 95 research papers focused on APT detection methodologies.

Fig. 3 shows the number of scientific papers proposing strategies that are able to detect C&C activity. According to the findings, 38 out of 95 APT detection solutions presented in papers were able to detect C&C activity. The remaining 57 APT detection techniques either do not detect C&C channels or do not indicate whether or not they detect C&C channels.

Table 4 shows which approaches are most frequently used for detecting beaconing during an APT attack in the selected papers. We classified the detection techniques that had been applied by researchers into four classes: behavior-based, signature-based, graph-based and machine or deep learning-based detection techniques.

Table 2
Artificial intelligence methods used to detect APT/beaconing.

Artificial Intelligence Algorithm	Ref.	Freq.	Artificial Intelligence Algorithm	Ref.	Freq.
Bayesian probabilistic	(Vert et al., 2018)	1	CNN+LSTM	(Ren et al., 2020; Highnam et al., 2021; Sivaguru et al., 2020; Tong et al., 2019; Ren et al., 2019; Dijk, 2021; Niu et al., 2022)	7
Bagging classification	(Nuojua et al., 2017)	1	K-Medoids	(Manzoor et al., 2016)	1
Categorical anomaly detection	(Berrada et al., 2020)	1	KNN	(Siddiqui et al., 2016; I. Ghafir et al., 2018; Zimba et al., 2020; Lu et al., 2019; Lu et al., 2016; de Vries et al., 2012; Nuojua et al., 2017; Shenwen et al., 2015)	8
CART	(Sharma et al., 2017; Barceló-Rico et al., 2016)	2	LinearSVM	(I. Ghafir et al., 2018)	1
Decision Tree	(I. Ghafir et al., 2018; Zhao et al., 2015; Moon et al., 2017; Barceló-Rico et al., 2016; Chu et al., 2019; D. Yan et al., 2019)	6	Logistic regression	(Sexton et al., 2016)	1
ELM	(Shi et al., 2018)	1	mSVMs	(Sharma et al., 2017)	1
EnseMLe	(I. Ghafir et al., 2018)	1	Naïve Bayes	(Chandran et al., 2015; Sexton et al., 2016; Nuojua et al., 2017; Debatty et al., 2018; Chandra et al., 2016; Chu et al., 2019)	6
FCM clustering	(Ge et al., 2016)	1	NN or DNN	(Yan et al., 2020; Nuojua et al., 2017; Debatty et al., 2018; Chu et al., 2019; Zhou et al., 2019; Abdullayeva, 2021)	6
Fuzzy means	(Lu et al., 2019; Lu et al., 2016; Ng and Bakhtiarib, 2016)	3	One-classSVMs	(Schindler, 2018; Chen et al., 2020; Dijk, 2021)	3
GBDT	(Lu et al., 2019; Lu et al., 2016)	2	Q-learning + greedy policy	(Xiao et al., 2018)	1
GP	(Sharma et al., 2017; Barceló-Rico et al., 2016)	2	Random Forest	(Hu et al., 2016; Chandran et al., 2015; Lamprakis et al., 2017; Barceló-Rico et al., 2016; Cho and Nam, 2019; Laurenza et al., 2017; D. Yan et al., 2019; Känzig et al., 2019; Lu et al., 2017; Niu et al., 2021)	10
GAF	(Niu et al., 2017)	1	SNN	(Zimba et al., 2020; Bodström and Hämäläinen, 2019)	2
Graph isomorphism algorithms-based	(Wang et al., 2014)	1	SVM	(Lu et al., 2019; Lu et al., 2016; Sexton et al., 2016; Barceló-Rico et al., 2016; J. Choi et al., 2015; Nuojua et al., 2017; Shan-Shan and Ya-Bin, 2017; Chu et al., 2019; D. Yan et al., 2019; Kondo and Sato, 2007)	10
Heuristic anomaly detection	(Bencsáth et al., 2012; Yu et al., 2019)	2	K-means	(Liu et al., 2019; Borchani, 2020; de Vries et al., 2012; G. Yan et al., 2019)	4
Hidden markov model	(Ghafir et al., 2019; Sengupta et al., 2019; Shan-Shan and Ya-Bin, 2017)	3	PGM + FG	(Cao, 2019)	1
Disciple multi-strategy learning approach	(Tecuici et al., 2018)	1	Canopy	(G. Yan et al., 2019)	1
Hierarchical clustering	(Balduzzi et al., 2013)	1	FP-Growth algorithm	(Lee et al., 2017)	1

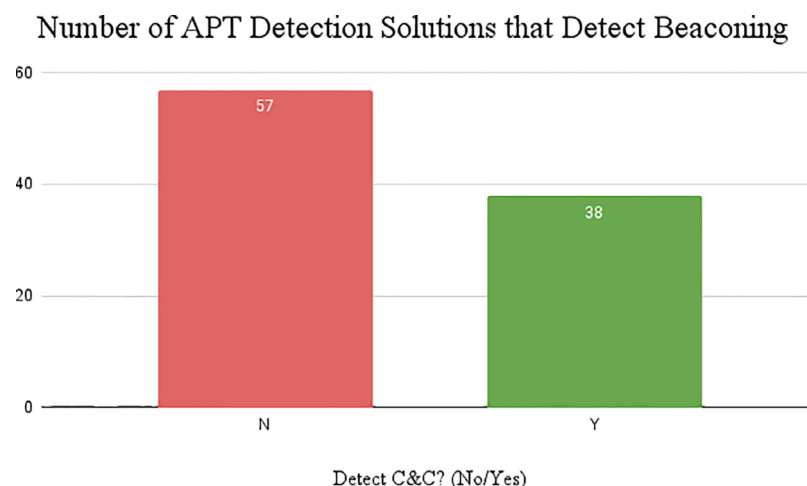


Fig. 3. Number of Scientific papers proposing APT detection techniques that detect beaconing.

Table 3
Beaconing Detection Techniques proposed by Scientific Papers.

Beaconing Detection Technique	Description	Ref.	Freq.	Percentage
Behavior based/network based detection	This method is based on host behavior or anomalous network traffic, such as excessive network latency, large volumes of traffic, traffic on unusual ports and anomalous system behaviors. This strategy is designed to notice any divergence from benign activity or any resemblance to C&C activity.	(Apruzzese et al., 2017; Richer, 2017; Chen et al., 2020; Vishvakarma et al., 2020; Liu et al., 2019; Jin et al., 2019; Seo and Lee, 2018; Ben-Asher et al., 2016; Borchani, 2020; Fedynyshyn et al., 2011; Jiang et al., 2019)	11	40.74%
Deep learning-based detection	This method utilizes various deep learning algorithms (particularly neural networks such as CNN, ANN, LSTM, etc.) to analyze time and space similarities to derive network traffic data. This approach entails converting network traffic into a grayscale picture or feature vector and passing it to a neural network model, where distinct characteristics and patterns are extracted from space and time dimensions, and network traffic characteristics are automatically learned.	(Ren et al., 2020; Highnam et al., 2021; Sivaguru et al., 2020; Zhou et al., 2019; Tong et al., 2019; Vinayakumar et al., 2019; Ren et al., 2019)	7	25.93%
Machine learning-based detection	Detection method based on finding common features and correlating different malware activities using different machine learning algorithms such as SVM, Random Forest, Decision Tree, etc.	(Oprea et al., 2018; D. Yan et al., 2019; Yu et al., 2019; Känzig et al., 2019; Y. Li et al., 2019; Lu et al., 2017; Kondo and Sato, 2007)	7	25.93%
Graph-based detection	By introducing linkages (or edges) between related anomaly patterns, graphs naturally describe interconnections. The associated patterns of long-range relationships are captured by the many pathways that run between them. A graph representation also allows for the addition of node and edge attributes/types, making it easier to describe large datasets.	(Tran et al., 2019)	1	3.70%
Signature-based detection	This method detects anomalous activities based on predefined patterns and signatures retrieved from well-known C2 activities. Common methods include regular expressions, whitelists (or blacklists) and N-gram models.	(Menon, 2019)	1	3.70%

Table 4
APT Detection Techniques that detect beaconing.

C2 detection technique	Ref.	Freq.	Percentage
Behavior-based and network-based detection	(Hu et al., 2016; Yan et al., 2020; Baksi and ; Upadhyaya, 2016; Atapour et al., 2018; Shenwen et al., 2015; Brogi and Tong, 2016; Su et al., 2022; Vance, 2014; Ng and Bakhtiarib, 2016; Lee et al., 2016; Niu et al., 2017; Kim et al., 2018; G. Yan et al., 2019; Wang et al., 2016)	14	36.8%
Signature-based detection	(Ghafir et al., 2019; I. Ghafir et al., 2018; Bhatt et al., 2014; Ghafir et al., 2017; Zhao et al., 2015; de Vries et al., 2012; Nuojua et al., 2017; Cho and Nam, 2019; Liu et al., 2012; Ghafir and Prenosil, 2016; Bencsáth et al., 2012; Moothedath et al., 2020; Najafi et al., 2021)	13	34.2%
Graph-based detection	(Manzoor et al., 2016; Zimba et al., 2020; Milajerdi et al., 2019; Lamprakis et al., 2017; Debatty et al., 2018; Hu et al., 2016)	6	15.8%
Machine or deep learning-based detection	(Lu et al., 2016; Chandran et al., 2015; Stoleriu et al., 2021; Niu et al., 2021; Dijk, 2021)	5	13.2%

Many research studies have employed a behavior/network-based detection approach for identifying APT beaconing activity. In this approach, network records are processed to identify the possible malicious source and destination pairs. The network activity of these pairs is taken within a specific time interval and converted from the time domain to the frequency domain for analysis. This is done because the analysis is focused on the behavior (pattern) rather than a particular event in the timeline. Finally, this processed network behavior is used to identify potential candidate frequencies and periodicities for beaconing operations. Examples of network behavior that can be analyzed based on its frequency include, but are not limited to, session count (in/out), MAC modulation, count of IP and ARP modulation happened, and the number of IP addresses belonging to the same destination (Moon et al., 2017). Thereby, this approach employs flow-based and statistical

measures to monitor, analyze and detect non-signature malicious traffic. Sketch-based estimations can then be applied to aggregated traffic for more accurate detection by computing and setting standard statistical measurements for known normal and abnormal network traffic (Singh et al., 2019). On the other hand, in signature-based detection methods, signatures extracted and gathered from actual reported APT beaconing assaults are utilized as solid evidence for identifying APT beaconing attacks (Ghafir et al., 2019, 2018). This is done by matching DNS logs to signatures acquired from actual C&C attack domains. If a match is found, the domain is identified as malicious; otherwise, it is benign. However, in many cases, the domains related to APT beaconing attacks are unknown and have similar characteristics of benign domains; hence, cannot be easily detected using this method. Similarly, graph-based detection methods identify the APT beaconing attack by evaluating

initial DNS requests and subsequent communications between internal and external hosts, then calculating and depicting a change in the number of communications to external hosts using graphs (Manzoor et al., 2016; Debatty et al., 2018). However, in this approach, the process of building the graphs is computationally intensive and requires a significant amount of time.

As discussed previously, beacons transmit signals to C&C servers regularly during the APT lifecycle. To detect beaconing, a security solution could look for patterns in communication time, such as GET and POST requests. While malware uses jitter (randomization) to disguise itself, it still creates a pattern that is easily detectable, especially by machine-learning based detection methods (Niu et al., 2021). Machine learning models, such as SVM and Random Forest, can be used to detect command and control communication by training the model on a large dataset of C2 attacks' features (e.g. features extracted from web proxy logs) to proactively detect external network connections resulting from malware communication (Oprea et al., 2018). Details about these datasets are provided in Section 4.3. Both behavior/network-based and machine learning-based detection methods strive to understand behavior, making them similar in that regard. However, unlike behavior-based learning, the process of learning behavior in machine learning is automated. Causing machine learning-based detection approaches to outperforms behavior-based methods in the process of learning behavior.

In this review, the most frequent technique used in APT detection systems that detect beaconing activity are behavior-based and network-based detection, closely followed by signature-based detection. These two detection technologies are backwards compatible, according to academic studies based on a variety of tests. Malware detection based on signatures is used to identify "known" malware. Unfortunately, signature-based systems are unable to detect novel versions of dangerous code. Only behavioral and network-based analysis can differentiate these newly revealed kinds of malware from innocuous APT command and control activities.

The 'Data Exfiltration' stage is the final stage in the APT life cycle. It is the act of stealing private, potentially valuable, data from a network and sending it to one or more external systems controlled by the attacker. Data Exfiltration is the stealthy action where the attacker exfiltrates the collected data to their command-and-control servers after gaining access, through establishing the C&C communication, to the information they are looking for. The data is usually exfiltrated at a very low transmission rate, unless the attacker is able to send them all at once and sees no benefit in remaining in the target system (Nar and Sastry, 2018). In order to avoid detection, the files could be reformatted, encrypted, or attached to other files before being exfiltrated. In the case of Duqu attack (Chien et al., 2012), the collected data were exfiltrated as JPEG files.

The findings of the study indicate that a number of research papers proposed one of the security protection mechanisms in APT detection solutions. That mechanism is one of the APT stages which is the capability to identify the data exfiltration. For example, (Sigholm and Bang, 2013) developed a method for preventing data leaks that makes use of the DLP algorithm to identify breaches and generates "fingerprints" based on the characteristics of each data transmission. Different sorts of information, including the destination and the hash file containing the sensitive information, may represent fingerprints. The information from the database containing the fingerprints was afterwards utilized by external sensors that monitor internet traffic to follow the path of the leaked data by searching for matching. Another research article (Zimba et al., 2020), discussed using semi-supervised learning approach based on an enhanced SNN-based clustering algorithm. By modeling the targeted network as a small-world network

model and the evolving APT-AN as a scale-free network model, the detection method is able to identify the data exfiltration stage. Nevertheless, many publications, like (Shi et al., 2018; Dijk, 2021; de Vries et al., 2012; Barceló-Rico et al., 2016; Lamprakis et al., 2017; Cao, 2019; Ghafir et al., 2018), were able to detect APT malware activities that involve data exfiltration, using artificial intelligence, machine learning, or deep learning methods. These methods concatenate domain knowledge, knowledge of previous attacks, and real-time observations from security monitors to detect data exfiltration. For instance, the AI model proposed by (Dijk, 2021) consists of a one class state vector machine, a stacked auto encoder, and a recurrent neural network. By examining the payload of the network traffic flow, flow-based deep packet inspection AI model can discover data breaches. Another publication (Barceló-Rico et al., 2016), developed a machine learning model by training it on both labelled and unlabelled anomalous set of traffic data. Genetic programming, decision trees, and support vector machines were the three computational intelligence techniques employed to train the classifiers. The outcomes demonstrate their potential capability for stopping APTs and finding data leaks. For the data exfiltration prediction during the APT attack, Ghafir et al. (2019) developed a probabilistic IDS. The suggested method consists of two primary stages: the first stage involves reconstructing the attack scenario using a correlation framework, and the second stage involves decoding the assault using an HMM. Besides these data exfiltration detection techniques, Marchetti et al. (2016) detected and classified suspicious hosts potentially implicated in APT-related data exfiltration. The hosts were classified according to the suspiciousness score assigned to each internal host. The normalized feature vectors of the internal hosts were used to create the suspiciousness scores using a statistical approach (likelihood algorithm).

4.3. Datasets/data sources and their characteristics (RQ3)

In this section, we present a detailed analysis of the datasets and data sources used in the APT detection methods and solutions proposed by the researchers.

Over the years, detecting APTs has remained a difficult task. The creation of a credible benchmark dataset for training and testing suggested techniques is an unavoidable step in APT detection research. Authors typically use realistic, synthetic or semi-synthetic datasets to assess proposed APT/beaconing detection systems (Messoud et al., 2016).

As mentioned above, there are three categories of datasets used to test APT detection solutions: realistic, synthetic, and semi-synthetic (Stojanović et al., 2020):

Realistic: a collection of data collected or created from real-world sources. This type of data allows for testing in real-world conditions (Antonopoulos et al., 2009). However, this form of dataset has several downsides: it is not easily scalable in terms of user input, the data gathered from office PCs are subject to privacy concerns, and there's a risk of the attack simulation harming the production system (Koroniatis et al., 2019). Furthermore, due to the nature of cyber attacks, the information must be updated on a regular basis, since attacks become more sophisticated over time (Koroniatis et al., 2019).

Synthetic: a collection of data that was generated artificially, rather than by actual events. Synthetic data is generated algorithmically and used as a stand-in for production or operational test datasets, mathematical model validation, and, increasingly, machine learning model training. This type of dataset has control over the data and the network setup, as the network can be set up in the desired manner with the preferred properties (Alshamrani et al., 2019). However, the problem with this type of data is that it takes the testing process out of context, removing all unknown properties and false alarms raised from real network

Dataset Type proposed by Scientific Papers

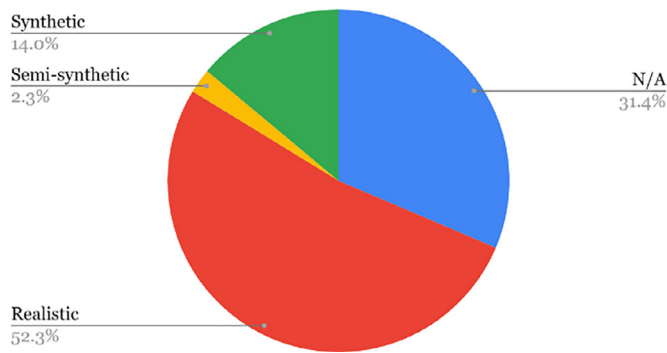


Fig. 4. Dataset creation type in scientific papers.

Dataset Creation Type in Scientific Papers

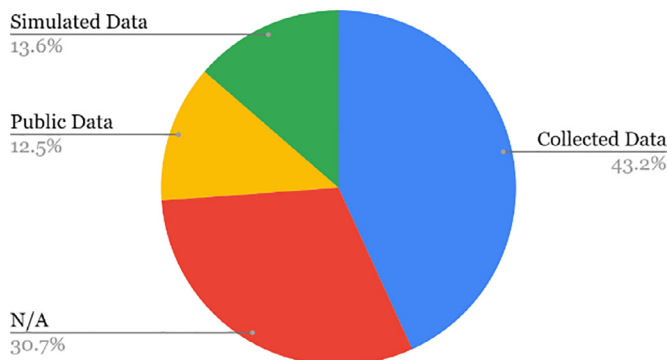


Fig. 5. Dataset type in scientific papers.

noise. A complete absence of noise presents several drawbacks: simulated attacks are oversimplified, potentially leading to unrealistically good detection results that are only valid for the test dataset, and would not stand up in a real-world context. Since attackers usually generate noise to stay undetected, this option is quite implausible (Alshamrani et al., 2019).

Semi-synthetic: a combination of realistic and synthetic datasets. Like synthetic datasets, the disadvantage of a semi-synthetic dataset is the potential failure of the detection methods in real-world conditions due to the use of an over-simplified dataset (Skopik et al., 2014). Moreover, the resulting dataset might follow an insufficiently accurate synthetic user model. However, the advantages of this type of dataset are that it is much less costly to create, and it is more easily scalable and adaptable to different scenarios (Skopik et al., 2014).

Fig. 5 illustrates the percentage of realistic, synthetic and semi-synthetic datasets used in the selected papers. Our analysis shows that, with a percentage of 52.3%, the most frequent dataset type used is the realistic type. Synthetic datasets were used in 14% of cases. However, only 2.3% of research papers used semi-synthetic datasets, in which part of the data was generated and another part was collected. For the remaining 31.4% of the datasets, the data description and type were not specified.

Fig. 4 presents the sources of the data in the datasets used in the collected research. It was found that 13.6% of datasets were created from simulated data, 43.2% were sourced from data that was collected from a given network, and 12.5% of the datasets were

created from publicly available datasets. In the remaining 30.7% of datasets, the data creation type was not specified.

Table 5 presents the publicly available datasets used in the collected research papers, outlining their characteristics to construct APT/beaconing detection systems. Because APTs are a type of intrusion, the properties developed for various techniques used by intrusion detection systems (IDSs) can also be used to detect advanced persistent threats. Overall, we identified ten different public datasets used in IDSs and supported by many APT/beaconing detection systems. Even though several other datasets were identified in this RQ, the DARPA dataset was used in the majority of the studies.

In addition, several realistic datasets were created, collected from traffic monitoring. For example, in Chandran et al. (2015), Brogi and Tong (2016), Moon et al. (2017), Wang et al. (2014), Moothedath et al. (2020), the researchers monitored a system for months, extracting the feature first, then launching known APT malware from an open malware site, and lastly extracting the feature again. The datasets include information about CPU usage, memory usage of the system, open ports, the number of files in the system32 folder, and domain names. Ge et al. (2016) constructed a dataset by analyzing memory images from the cloud. They developed a classifier to extract feature information and store data in the database, such as DLL file names and numbers, special API function names and numbers, special registered information and system paths in the process memory.

Moreover, some of the studies (Ghafir et al., 2017; Borchani, 2020; Liu et al., 2012, 2013; Rass et al., 2017; Ghafir et al., 2018; Balduzzi et al., 2013), used Bro passive, which is an open-source software for analyzing traffic, along with other tools to capture and monitor traffic. These tools are typically used to monitor security by carefully examining all traffic on a given link for signals of suspicious activity. The datasets include log files, which are high-level records of network activity. These logs contain not only a complete record of every connection made over the network, but also application-layer transcripts such as all HTTP sessions with their requested key headers, URIs, MIME types, and server answers such as DNS queries with responses (Ghafir et al., 2019), and much more.

Moreover, Ghafir and Prenosil (2016) collected different blacklists from several resources, such as lists of exploited domain names (FQDNS), blacklists of malicious domains, blacklists of file hashes, blacklists of SSL certificates, blacklists of C&C servers, and lists of all current or server IP addresses. These blacklists were obtained from different resources such as: Abuse.ch., www.mandiant.com, www.malware-domains.com, www.malwaredomainlist.com and www.blade-defender.org. In research studies (Hu et al., 2016; Li et al., 2018), a blacklist consisting of known DGA domains was used.

Research papers (Bencs  th et al., 2012; Marchetti et al., 2016; Vance, 2014; Maccari et al., 2018) analyzed network traffic with flow collectors to capture logs of network traffic meta-data including source and destination IP address, source and destination ports, time the connection was established, end time, and number of packets and bytes transferred. Researchers Yan et al. (2020), Tran et al. (2019), Liu et al. (2019), Seo and Lee (2018), Shi et al. (2018), Niu et al. (2017), Chandra et al. (2016), Yan et al. (2019) constructed datasets of DNS requests each day from a regional base station. The information included user, source IP, destination IP, country flag, domain name, request time and response time. These studies focused on collecting legitimate and APT malicious domains (Cho and Nam, 2019; Zhou et al., 2019) collected C&C server domains). Furthermore, (Su et al., 2022) created a dataset of segments and documents that may contain malicious forms, such as HTML text files, documents (doc, xls, pdf, etc.), ex-

Table 5
Publicly Available Datasets used in APT/Beaconing Detection.

Ref	Public Dataset	Characteristics
(Siddiqui et al., 2016)	PREDICT	Normal and non-malicious data is obtained from PREDICT internet dataset repository under the category of "DARPA Scalable Network Monitoring (SNM) Program Traffic". The PREDICT dataset was filtered to extract normal packet flows. The APT dataset was combined with these normal flows to generate a dataset mimicking the mechanism of an APT attack. The total size of DARPA PCAP files is 6 TB and contains HTTP, SMTP and DNS data. The dataset contains data about the following: 1. Authentication 2. Processes 3. DNS 4. Network Flow 5. Red Team They simulated several APT attacks that contained whole or parts of the attack and provided two months of anonymized DNS records collected from a large site. In this dataset, real attacks might exist, because the data came from a real site rather than network traffic generators.
(Zimba et al., 2020; Wang et al., 2016)	Dataset of the Los Alamos National Security Laboratory	Contagio is a public collection of the latest malware samples, threats, observations, and analyses. It contains ATP malware names and details. The normal and background traffic come from the university gateway lab server. The traffic protocol includes UDP/TCP/HTTP/SMTP/DNS and others. The APT 1 malware considered here consists of 197 programs in 37 sub-families. Each program was successfully disassembled using IDA-Pro. This paper focused on designing classifiers based on disassembled executables. In addition to the APT malware, a sample of 4622 non-APT disassembled programs is also used.
(Siddiqui et al., 2016; Lu et al., 2016; Lu et al., 2019)	Public Contagio Malware Dataset	The benign programs were taken from a program analysis tool and repository at Los Alamos National Laboratory called CodeVision. DARPA
(Sexton et al., 2016)	The benign programs were taken from a program analysis tool and repository at Los Alamos National Laboratory called CodeVision.	Public data collection from DARPA representing APT attack scenarios, each consisting of several days of processes and net flow activities in a DARPA evaluation of provenance-tracking systems. Runs on Windows, FreeBSD, Linux and Android.
(Siddiqui et al., 2016; Milajerdi et al., 2019; Niu et al., 2017; Xiong et al., 2020) (Chu et al., 2019)	NSL-KDD	The predecessor of the NSL-KDD dataset ² was an improved version of KDD 99 (based on a database established by DARPA in 1999), which had redundant data removed and overcame the classifier recurring records problem that tended to affect learning performance. 7 features taken from raw network data and 9 features retrieved from log files are included in the dataset, which was developed in 2016 in an emulated network environment. The NSL-KDD dataset has basic feature information including time and traffic. It stores packet-based network communications as well as log files from hosts. Backdoors, DoS, exploits, generic, reconnaissance, shellcode, and worms are among the attack families included in this dataset. Each NSL-KDD network data record has 38 digital type attribute features, as well as three-character type attribute features including protocol type, service, and flag. In 2011, botnet traffic was intercepted at the CTU University in the Czech Republic. The CTU-13 dataset consists of thirteen different botnet samples. They ran a distinct malware on each botnet sample, each of which used various protocols and did various actions. Each sample was recorded in a pcap file that included all three types of traffic packets. Other types of data, such as NetFlows and WebLogs, were extracted from these pcap files. The ISOT dataset is a collection of malicious and non-malicious datasets that are publicly available. Behavioral biometric datasets Botnet and ransomware detection datasets Cloud security datasets Fake news detection datasets Stylometry authentication datasets
(Chen et al., 2020)	CTU-13 ³	The CCC DATASET is a list of hash digests for gathered malware samples, packet traces, and malware collection logs obtained by the Cyber Clean Center's server-side, high-interaction distributed honeypots. The dataset includes HTTP and IRC conversations. This dataset was intended to be shared with the scientific community in order to provide a starting point for anybody interested in using Machine Learning for Malware Analysis. The following samples make up the gathered dataset: Crypto 2024 Samples Locker 434 Samples Zeus 2014 Samples APT1 292 Samples
(Richer, 2017)	ISOT ⁴	
(Lu et al., 2017)	CCC DATASET 2008, 2009, and 2010 ⁵	
(Abdullayeva, 2021)	MalwareTrainingSets ⁶	

¹<https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset>.

²<http://www.unb.ca/cic/datasets/ns1.html>.

³<https://www.stratosphereips.org/datasets-ctu13>.

⁴<https://www.uvic.ca/engineering/ece/isot/datasets/>.

⁵<http://www.iwsec.org/mws/2014/about.html>.

⁶<https://marcoramilli.com/2016/12/16/malware-training-sets-a-machine-learning-dataset-for-everyone>.

executable files, or sets of packets sent to an application installed in the host.

Using a different approach, synthetic and semi-synthetic attack data were simulated and collected in datasets by Schindler (2018), Ghafir et al., (2018), Lv et al. (2019), Ghafir et al. (2019), Atapour et al. (2018), Manzoor et al. (2016), Friedberg et al. (2015), Sharma et al. (2017) and Bencsáth et al. (2012). They customized and developed attack scenarios to simulate different APT attack phases across several machines, generating data from multiple detection sensors and concluding in successful data theft. Moreover, the authors in Manzoor et al. (2016) assisted in the construction of the datasets used in the present study (i.e., collecting and pre-processing system-call traces for benign and malicious scenarios). The dataset included system call flow-graphs from typical browser activity, as well as numerous simulated abnormal attack scenarios. The collection is composed of flow-graphs from one attack and five benign situations. Normal internet activities, such as viewing YouTube, downloading files, surfing CNN.com, checking Gmail, and playing a video game, are included in the benign scenarios. From the beginning of a task until its completion, all system calls on the machine were tracked and used to create the flow-graph for that task. Three datasets were created from the flow-graphs using a semi-synthetic data production method devised by Sharma et al. (2017). It's a hybrid strategy that combines the development of synthetic data with the collection of useful log data in the virtual machine environment by recording data for assessment from real systems. According to the authors, employing a virtual environment improves data quality because the gathered records are extremely similar to those gathered in a productive context, but without the potential for noise. The objectives of this work were to generate network flow, system events, and operational statistics for a complex ICT services scenario.

4.4. Strengths and limitations (RQ4)

To answer RQ4, the strengths and challenges of the APT/beaconing detection strategies and approaches presented in the previous subsections are addressed in this section.

The strengths and limitations of the selected research papers are presented in Table 6. Please be aware that the papers listed in the table do not represent all of the published papers selected for analysis. In addition, many scientific papers proposed more than one strength or limitation. In our research, we discovered that the majority of APT/beaconing detection solution weaknesses fall into the following categories: complex or expensive implementation, inability to detect all phases of the APT lifecycle, inability to detect some types of APT/beaconing attack, low detection rate and high cost of performance/resource overheads. We believe that these limitations will encourage researchers to consider those specific areas in their future work. On the other hand, the proposed approaches' strengths mostly revolved around detecting APT/beaconing with high accuracy, allowing high performance and efficiency in recognizing APT/beaconing attacks, identifying unknown attacks and enabling early and timely detection/prediction and high detection speed. Furthermore, scientific publications focused on each solution's capacity to identify and forecast attacks in realtime.

4.5. APT/ beaconing detection vendor projects

This section discusses different anti-APT/beaconing software projects developed to detect APT or beaconing activities.

In this review, we gathered 31 cybersecurity software projects that could detect APT or beaconing attack activities. Fig. 6 below

License Type proposed by Cybersecurity Vendor Projects

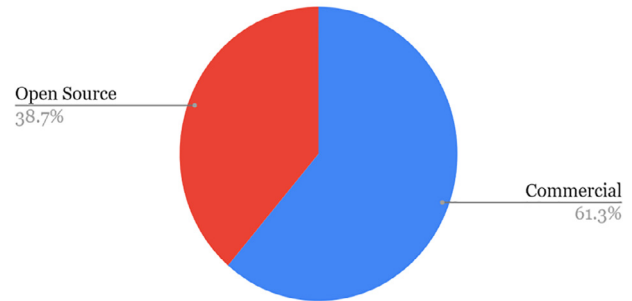


Fig. 6. Software license type proposed by cybersecurity vendor projects.

shows the percentage of anti-APT/beaconing software projects developed as open-source products versus those developed as commercial products. The review illustrates that, with a percentage of 61.3%, the most frequent anti-APT/beaconing software license type is the commercial product type, while the remaining 38.7% of the projects were open-source products.

According to our findings, we identified several APT/beaconing detection approaches adopted by anti-APT/beaconing software projects. According to Table 7, most projects use the **Network analysis approach**. This approach tracks suspicious behavior by monitoring the network/data logs and using real-time correlation policies to detect APT or beaconing activity (Lu et al., 2019). However, a large proportion of projects do not specify the detection techniques used to detect targeted APT or beaconing attack activities. Other detection approaches, such as artificial intelligence-based detection, whitelisting, sandboxes, and multi-layered APT/beaconing detection approaches are also utilized by these projects. The difference between these approaches is as follows:

Artificial intelligence-based detection: This approach claims to use artificial intelligence and machine learning techniques to detect APT/beaconing attacks (Machine Learning in Cybersecurity | Kaspersky 2022; "RSA NetWitness Platform Documentation - RSA Link 2021). Because AI models are trained by supplying them with a dataset of known attack behavior, the detection results of this technique are usually biased towards identifying known attack patterns.

Whitelisting: A whitelist is a list of authorized entities in general. Whitelisting is a cybersecurity approach used in APT or beaconing detection that authorizes a list of email addresses, IP addresses, or domain names while rejecting all others ("Configuring white list mode 2022). This technique manages domains that can be accessed from a given network and programs that network users can install. This technique can be used to determine whether an apt/beaconing attack used a benign domain.

Sandbox: This approach employs a layer of network security defense against APTs and beaconing. Its dynamic testing detects malware by executing (or activating) code in a secure and isolated environment and analyzing the malware code's behavior and output ("Barracuda CloudGen Firewall | Barracuda Networks 2022; Advanced Malware Detection - Advanced Threat Protection | Forcepoint 2022). In other words, a sandbox protocol isolates a certain application from the rest of the system. There, the suspicious object's behavior is evaluated, and other systems are protected from its harmful effect. If the suspicious software runs malicious code, only the protected, segregated sandbox is impacted.

Multi-layered approach: This method computes and analyzes numerous network traffic events in order to detect anomalous indications and behaviors and draw conclusions about whether or not APT/beaconing is present in the system. This strategy

Table 6
Strengths and weaknesses of scientific papers.

Strengths	Ref	Limitation	Ref.
High detection rate / accuracy	(Bryant and Saiedian, 2017; Chandran et al., 2015; Schindler, 2018; Hu et al., 2016; Ghafir et al., 2019; Yan et al., 2020; Ge et al., 2016; Sharma et al., 2017; Brogi and Tong, 2016; Zhao et al., 2015; Vance, 2014; Lamprakis et al., 2017; Haopu, 2016; Shan-Shan and Ya-Bin, 2017; Bodström and Hämäläinen, 2019; Liu et al., 2012; Rubio et al., 2017; Cho and Nam, 2019; Cui et al., 2019; Chu et al., 2019; Sigholm and Bang, 2013; Laurenza et al., 2017; Xiong et al., 2020; Chandra et al., 2016; Känzig et al., 2019; Seo and Lee, 2018; Fedynyshyn et al., 2011; Kondo and Sato, 2007; Jiang et al., 2019; Abdullayeva, 2021)	Slow learning time	(Xiao et al., 2018)
Early and timely attack prediction / detection	(M. Marchetti et al., 2016; I. Ghafir et al., 2018; Yan et al., 2020; Moon et al., 2017; Manzoor et al., 2016; Saud and Islam, 2015; Su et al., 2022; Berrada et al., 2020; I. Ghafir et al., 2018)	Low detection rate / high false positive / low accuracy	(I. Ghafir et al., 2018; Friedberg et al., 2015; Moon et al., 2017; Brogi and Tong, 2016; Ghafir et al., 2017; Bhatt et al., 2014; de Vries et al., 2012; J. Choi et al., 2015; Highnam et al., 2021; Vishvakarma et al., 2020; Najafi et al., 2021)
Realtime network analysis / realtime detection and analysis;	(M. Marchetti et al., 2016; I. Ghafir et al., 2018; Cao, 2019; Milajerdi et al., 2019; Ghafir et al., 2017; Ghafir and Prenosil, 2016; Shi et al., 2018; Apruzzese et al., 2017; Ren et al., 2020; Jin et al., 2019; Liu et al., 2019)	Not able to detect all stages of APT lifecycle	(Ghafir et al., 2019; Baksi and Upadhyaya, 2016; Lamprakis et al., 2017; I. Ghafir et al., 2018; Balduzzi et al., 2013; Chandra et al., 2016; G. Yan et al., 2019)
High performance	(Huang and Zhu, 2019; Siddiqui et al., 2016; Lv et al., 2019; Shenwen et al., 2015; Mirza et al., 2014; Wang et al., 2016; Bhatt et al., 2014; Xiao et al., 2018; Hu et al., 2017; Beuhring and Salous, 2014; Bodström and Hämäläinen, 2019; Debatty et al., 2018; Hu et al., 2015; Balduzzi et al., 2013; G. Yan et al., 2019; Känzig et al., 2019; Ren et al., 2019; Fedynyshyn et al., 2011; Lu et al., 2017)	Cannot achieve realtime detection	(Cao, 2019; Balduzzi et al., 2013; Ren et al., 2019)
:	(Ge et al., 2016; J. Choi et al., 2015; Lee et al., 2016; Liu et al., 2013; Chu et al., 2019; Xiong et al., 2020; Richer, 2017)	Some types of APT/beaconing cannot be detected	(Lu et al., 2016; Wang et al., 2016; Zhao et al., 2015; Wang et al., 2014; Laurenza et al., 2017; Tong et al., 2019; Liu et al., 2019; Abdullayeva, 2021)
Flexibility to be integrated into detection systems	(Schindler, 2018; Yan et al., 2020; Känzig et al., 2019)	Reduce system efficiency	(Jia et al., 2015)
Identifies different characteristics / behavior of APT/beaconing attack	(Schindler, 2018; Atapour et al., 2018; Sexton et al., 2016; Jia et al., 2015; Lee et al., 2017; Vishvakarma et al., 2020)	Complex or expensive implementation	(Li et al., 2018; Kim and Park, 2014; de Vries et al., 2012; Beuhring and Salous, 2014; Bodström and Hämäläinen, 2019; Tecuci et al., 2018; Kim et al., 2018; Debatty et al., 2018; Jin et al., 2019)
Efficiency	(Schindler, 2018; Lv et al., 2019; Yan et al., 2020; Lu et al., 2016; Mirza et al., 2014; J. Choi et al., 2015; Zhao et al., 2015; Vance, 2014; J. Choi et al., 2015; Niu et al., 2017; ; Xiong et al., 2020; Maccari et al., 2018; G. Yan et al., 2019; Oprea et al., 2018; Menon, 2019; Ren et al., 2019)	Time complexity	(Bodström and Hämäläinen, 2019; Liu et al., 2012)
Detects wide range of potential unknown APT/beaconing attacks	(Baksi and Upadhyaya, 2016; Lu et al., 2019; de Vries et al., 2012; ; Barceló-Rico et al., 2016; Nuojua et al., 2017; Tecuci et al., 2018; Vert et al., 2018; Y. Li et al., 2019; Bencsáth et al., 2012)	Detection system is not robust	(Berrada et al., 2020)
Resource usage remains consistent / minimizes resource cost / sustainability of system	(Zhao et al., 2015; Moothedath et al., 2020; Xiong et al., 2020)	Cost of performance / resource overheads	(Shenwen et al., 2015; Milajerdi et al., 2019; Cui et al., 2019; Barceló-Rico et al., 2016)
Detects different stages of APT lifecycle	(Zimba et al., 2020; Zhu and Rass, 2018)	High computational overhead	(Zimba et al., 2020; Lu et al., 2019; Milajerdi et al., 2019)
Fast learning speed	(Shi et al., 2018)	Takes a long time to detect/predict APT or beaconing attack	(Cho and Nam, 2019; Giura and Wang, 2012; Hu et al., 2017; Menon, 2019)
Analyzes large amounts of data	(Shenwen et al., 2015; F. Skopik et al., 2014; Bodström and Hämäläinen, 2018)	--	--

primarily employs a number of serial primary layers to detect APT or beaconing ("Symantec Endpoint Protection 12.1 Business Pack – Tecdeal 2022; Preventing Multi-layered Cybersecurity Threats, 2022; Advanced Malware Detection - Advanced Threat Protection | Forcepoint, 2022). For instance, the first layer detects

APT/beaconing attacks by analyzing abnormal connections, the second layer detects APT/beaconing attacks by analyzing and evaluating Suricata logs, the third layer detects APT/beaconing attacks by analyzing behavior profiles compiled from the first layer, and so on.

Table 7
APT/Beaconing Detection Techniques proposed by Projects.

APT/Beaconing Detection Approach	Project ID	Freq.	Percentage
Network analysis / anomaly detection / data logs analysis	P2, P3, P5, P6, P7, P8, P14, P22, P28, P4, P9, P11, P18	13	41.9%
Artificial intelligence detection	P1, P15, P26	3	9.7%
Whitelisting	P12	1	3.2%
Sandbox	P20, P24	2	6.5%
N/A	P10, P16, P17, P19, P21, P23, P25, P27, P31	9	29%
Multi-layered approach	P13, P29, P30	3	9.7%

5. Conclusion and future work

In this SLR, we analyzed and compared the techniques used in the current solutions for detecting APTs. We also studied and reviewed beaconing detection techniques, whether they occurred during APTs or not. We examined several artificial intelligence (AI) algorithms and data sources used by researchers. Finally, we presented the strengths and limitations of the proposed solutions. The following is a summary of our findings:

RQ1 identified and examined APT detection methodologies proposed by selected scientific papers. We found that the most frequently utilized APT detection techniques are network flow analysis-based, signature-based, graph-based, game-based, and correlation analysis detection methods. Additionally, we discussed the AI algorithms adopted by the APT and beaconing detection methods and solutions offered in the selected research articles. Most of the beaconing detection solutions leverage a combination of CNN and LSTM algorithms, while most of the APT detection solutions take advantage of the deep/machine learning algorithms KNN and SVM.

RQ2 summarized the beaconing detection techniques applied by 27 research papers that were unrelated to the APT lifecycle. We identified three approaches that were frequently used in beaconing detection solutions: behavior- and network-based detection strategies, machine learning methods and deep learning detection methods. RQ2 also determined detection techniques that focus on detecting beaconing during APT attacks. We found that 38 out of 95 research papers with findings related to APT detection methods were focused on detecting beaconing activities. The most frequently used approaches were behavior-based and network-based detection strategies and signature-based detection strategies.

RQ3 discussed the most frequently used data sources and datasets in the selected scientific papers. We categorized datasets into realistic, synthetic, and semi-synthetic types. We found that at 52.3%, realistic datasets were most frequently applied in the training and developing of APT/beaconing solutions in scientific papers, whereas synthetic and semi-synthetic datasets were used by only 14% and 2.3% of scientific papers, respectively. Furthermore, we discovered that the DARPA dataset is the most frequently utilized publicly available dataset by the majority of researchers.

RQ4 showed that the main limitations of APT/beaconing detection solutions occur in the form of implementation complexity, low accuracy, inability to detect all phases of the APT lifecycle, inability to detect some types of APT/beaconing attack, and high cost of performance/resource overheads. On the other hand, the main strengths found in the proposed detection methods are high accuracy, high performance and efficiency in recognizing APT/beaconing attacks, the ability to identify unknown attacks, early and timely detection/prediction and high detection speed. Furthermore, several scientific publications em-

phasized the solution's ability to detect and predict attacks in realtime.

When we analyzed anti-APT/beaconing software projects, we found that 61.3% of the projects were licensed as commercial products while the remaining 38.7% were licensed as open-source products. In addition, the most frequently used detection approach used by the projects was the network analysis detection approach, which was present in 41.9% of the selected projects.

Finally, we identified several potential future work prospects based on the results of the SLR. Given that many researchers did not appear to consider or outline the complexity of implementation, the performance, or the necessity of lowering overhead costs and resource consumption of APT/beaconing detection methodologies, these will be an important area for improvement. Furthermore, most publicly available datasets do not allow for testing at all stages of the APT attack lifecycle. As a result, creating representative datasets for APT testing can be another future direction for research, as it can improve the performance of APT/beaconing detection techniques.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Manar Abu Talib: Conceptualization, Validation, Funding acquisition, Writing – review & editing, Supervision. **Qassim Nasir:** Conceptualization, Resources, Project administration, Supervision, Writing – review & editing. **Ali Bou Nassif:** Conceptualization, Resources, Project administration, Supervision, Writing – review & editing. **Takua Mokhamed:** Methodology, Formal analysis, Data curation, Investigation, Resources, Visualization, Writing – original draft. **Nafisa Ahmed:** Methodology, Formal analysis, Data curation, Investigation, Resources, Visualization, Writing – original draft. **Bayan Mahfood:** Formal analysis, Data curation, Investigation, Visualization, Writing – original draft.

Acknowledgment

Dr. Manar Abu Talib and her co-authors would like to thank the University of Sharjah, OpenUAE Research and Development Group, Dubai Electricity and Water Authority (DEWA), Ioannis Vaxevas, Abdulla Algaoud, and Ahmed Al Ketbi for funding and contributions to this SLR research. We also appreciate the help of our research assistants in selecting, summarizing and interpreting research papers for this SLR review.

Appendix

Table 8

Selected Research Papers.

Title	Type	Year	Ref.	Title	Type	Year	Ref.
A novel kill-chain framework for remote security log analysis with SIEM software	Journal	2017	(Bryant and Saiedian, 2017)	A technology for detection of advanced persistent threat in networks and systems using a finite angular state velocity machine and vector mathematics	chapter	2018	(Vert et al., 2018)
Adaptive Strategic Cyber Defense for Advanced Persistent Threats in Critical Infrastructure Networks	Journal	2019	(Huang and Zhu, 2019)	A Model of APT Attack Defense Based on Cyber Threat Detection	Conference	2019	(Y. Li et al., 2019)
An Efficient Classification Model for Detecting Advanced Persistent Threat	Conference	2015	(Chandran et al., 2015)	Malicious domain name detection based on extreme machine learning	Journal	2017	(Shi et al., 2018)
Analysis of high volumes of network traffic for Advanced Persistent Threat detection	Journal	2016	(M. Marchetti et al., 2016)	A Method of Monitoring and Detecting APT Attacks Based on Unknown Domains	Journal	2019	(Cho and Nam, 2019)
Anomaly Detection in Log Data using Graph Databases and Machine Learning to Defend Advanced Persistent Threats	Article	2017	(Schindler, 2018)	General Sum Markov Games for Strategic Detection of Advanced Persistent Threats using Moving Target Defense in Cloud Networks	Conference	2019	(Sengupta et al., 2019)
BAYWATCH: Robust Beacons Detection to Identify Infected Hosts in Large-Scale Enterprise Networks	Conference	2016	(Hu et al., 2016)	Ontology modeling for APT attack detection in an IoT-based power system	Conference	2018	(Kim et al., 2018)
Countering Advanced Persistent Threats through Security Intelligence and Big Data Analytics	Conference	2016	(M. Marchetti et al., 2016)	Research of Snort Rule Extension and APT Detection Based on APT Network Behavior Analysis	Conference	2019	(Cui et al., 2019)
Defending against the Advanced Persistent Threat: An Optimal Control Approach	Journal	2018	(Li et al., 2018)	Graph-based APT detection	Conference	2018	(Debatty et al., 2018)
Detecting advanced persistent threats using fractal dimension-based machine learning classification	Conference	2016	(Siddiqui et al., 2016)	A novel search engine to uncover potential victims for apt investigations	Conference	2013	(Liu et al., 2013)
Detection of advanced persistent threat using machine-learning correlation analysis	Journal	2018	(I. Ghafir et al., 2018)	Detection and Classification of Advanced Persistent Threats and Attacks Using the Support Vector Machine	Journal	2019	(Chu et al., 2019)
Dynamic defense strategy against advanced persistent threat under heterogeneous networks	Journal	2019	(Lv et al., 2019)	Duqu: Analysis, detection, and lessons learned	Conference	2012	(Bencsáth et al., 2012)
Hidden Markov Models and Alert Correlations for the Prediction of Advanced Persistent Threats	Journal	2019	(Ghafir et al., 2019)	A Novel Method for Detecting APT Attacks by Using OODA Loop and Black Swan Theory	Conference	2018	(Bodström and Hämäläinen, 2018)
Discovering Suspicious APT Behaviors by Analyzing DNS Activities	Journal	2020	(Yan et al., 2020)	The APT Detection Method based on Attack Tree for SDN	Conference	2018	(Shan-Shan and Ya-Bin, 2018)
Kidemonas: The Silent Guardian	Article	2017	(Baksi and Upadhyaya, 2016)	Towards Offensive Cyber Counterintelligence: Adopting a Target-Centric View on Advanced Persistent Threats	Conference	2013	(Sigholm and Bang, 2013)
Modeling Advanced Persistent Threats to enhance anomaly detection techniques	Journal	2018	(Atapour et al., 2018)	Advanced Persistent Threat Detection Method Research Based on Relevant Algorithms to Artificial Immune System	Conference	2015	(Jia et al., 2015)
Modeling and detection of the multi-stages of Advanced Persistent Threats attacks based on semi-supervised learning and complex networks characteristics	Journal	2020	(Zimba et al., 2020)	Malware Triage Based on Static Features and Public APT Reports	Conference	2017	(Laurenza et al., 2017)

(continued on next page)

Table 8
(continued)

Title	Type	Year	Ref.	Title	Type	Year	Ref.
On Multi-Phase and Multi-Stage Game-Theoretic Modeling of Advanced Persistent Threats	Journal	2018	(Zhu and Rass, 2018)	A Game-Theoretic Approach for Dynamic Information Flow Tracking to Detect Multi-Stage Advanced Persistent Threats	Journal	2020	(Moothedath et al., 2020)
On Preempting Advanced Persistent Threats Using Probabilistic Graphical Models	Article	2019	(Cao, 2019)	Defending against advanced persistent threats using game-theory	Journal	2017	(Rass et al., 2017)
The Big Four - What we did wrong in Advanced Persistent Threat detection	Conference	2013	(Virvilis and Gritzalis, 2013)	Dynamic defense strategy against advanced persistent threat with insiders	Conference	2015	(Hu et al., 2015)
A study on cyber threat prediction based on intrusion detection event for APT attack detection	Journal	2012	(Kim and Park, 2014)	CONAN: A Practical Real-time APT Detection System with High Accuracy and Efficiency	Journal	2022	(Xiong et al., 2020)
Combating advanced persistent threats: From network event correlation to incident detection	Journal	2015	(Friedberg et al., 2015)	Detection: Definition of New Model to Reveal Advanced Persistent Threat	Conference	2018	(Maccari et al., 2018)
APT traffic detection based on time transform	Conference	2016	(Lu et al., 2016)	APT attack behavior pattern mining using the FP-growth algorithm	Conference	2017	(Lee et al., 2017)
A context-based detection framework for advanced persistent threats	Conference	2012	(Giura and Wang, 2012)	Disguised executable files in spear-phishing e-mails: Detecting the point of entry in advanced persistent threat	Conference	2018	(I. Ghafir et al., 2018)
A temporal correlation and traffic analysis approach for APT attacks detection	Journal	2017	(Lu et al., 2019)	Targeted attacks detection with SPUge	Conference	2013	(Balduzzi et al., 2013)
Study and research of APT detection technology based on big data processing architecture	Conference	2015	(Shenwen et al., 2015)	A practical approach to E-mail spam filters to protect data from advanced persistent threat	Conference	2016	(Chandra et al., 2016)
Anticipating Advanced Persistent Threat (APT) countermeasures using collaborative security mechanisms	Conference	2014	(Mirza et al., 2014)	AULD: Large Scale Suspicious DNS Activities Detection via Unsupervised Learning in Advanced Persistent Threats	Journal	2019	(G. Yan et al., 2019)
Holmes: real-time apt detection through correlation of suspicious information flows	Conference	2019	(Milajerdi et al., 2019)	Advanced Persistent Threat attack detection method in cloud computing based on autoencoder and softmax regression algorithm	Journal	2021	(Abdullayeva, 2021)
DTB-IDS: an intrusion detection system based on decision tree using behavior analysis for preventing APT attacks	Journal	2017	(Moon et al., 2017)	SIEMA: Bringing Advanced Analytics to Legacy Security Information and Event Management	Conference	2021	(Najafi et al., 2021)
Detection of command and control in advanced persistent threat based on independent access	Conference	2016	(Wang et al., 2016)	Cyber Attacks Detection Using Open Source ELK Stack	Conference	2021	(Stoleriu et al., 2021)
Fast memory-efficient anomaly detection in streaming heterogeneous graphs	Conference	2016	(Manzoor et al., 2016)	Malware on Internet of UAVs Detection Combining String Matching and Fourier Transformation	Journal	2021	(Niu et al., 2021)
Towards proactive detection of advanced persistent threat (APT) attacks using honeypots	Conference	2015	(Saud and Islam, 2015)	Detection of Advanced Persistent Threats using Artificial Intelligence for Deep Packet Inspection	Conference	2021	(Dijk, 2021)
An APT Trojans Detection Method for Cloud Computing Based on Memory Analysis and FCM	Conference	2016	(Ge et al., 2016)	MADE: Security Analytics for Enterprise Threat Detection	Journal	2018	(Oprea et al., 2018)
Malicious SSL Certificate Detection: A Step Towards Advanced Persistent Threat defense	Conference	2017	(Ghafir et al., 2017)	Identifying malicious hosts involved in periodic communications	Journal	2017	(Apruzzese et al., 2017)
DFA-AD: a distributed framework architecture for the detection of advanced persistent threats	Journal	2016	(Sharma et al., 2017)	Entropy-based Detection of Botnet Command and Control	Conference	2017	(Richer, 2017)

(continued on next page)

Table 8
(continued)

Title	Type	Year	Ref.	Title	Type	Year	Ref.
Towards a framework to detect multi-stage advanced persistent threats attacks	Conference	2014	(Bhatt et al., 2014)	A DGA domain names detection modeling method based on integrating an attention mechanism and deep neural network	Journal	2020	(Ren et al., 2020)
Subroutine based detection of APT malware	Journal	2016	(Sexton et al., 2016)	Real-Time Detection of Dictionary DGA Network Traffic using Deep Learning	Article	2020	(Highnam et al., 2021)
TerminAPTor: Highlighting Advanced Persistent Threats through Information Flow Tracking	Conference	2016	(Brogi and Tong, 2016)	Inline Detection of DGA Domains Using Side Information	Article	2020	(Sivaguru et al., 2020)
Ontology based APT attack behavior analysis in cloud computing	Conference	2015	(J. Choi et al., 2015)	A Novel Ensemble Anomaly based Approach for Command-and-Control Channel Detection	Conference	2020	(Chen et al., 2020)
A framework of apt detection based on dynamic analysis	Conference	2015	(Su et al., 2022)	CNN-based DGA Detection with High Coverage	Journal	2019	(Zhou et al., 2019)
Detecting APT malware infections based on malicious DNS and traffic analysis	Journal	2015	(Zhao et al., 2015)	D3N: DGA Detection with Deep-Learning Through NXDomain	Conference	2019	(Tong et al., 2019)
Attacker-Centric View of a Detection Game against Advanced Persistent Threats	Journal	2018	(Xiao et al., 2018)	Pontus: A Linguistics-based DGA Detection System	Journal	2019	(D. Yan et al., 2019)
Systems for Detecting Advanced Persistent Threats: A Development Roadmap Using Intelligent Data Analysis	Conference	2012	(de Vries et al., 2012)	Detection of Algorithmically Generated Domain Names in Botnets	Conference	2019	(Vishvakarma et al., 2020)
A Network Gene-Based Framework for Detecting Advanced Persistent Threats	Conference	2014	(Wang et al., 2014)	Improved DGA Domain Names Detection and Categorization Using Deep Learning Architectures with Classical Machine Learning Algorithms	Conference	2019	(Vinayakumar et al., 2019)
Flow based analysis of Advanced Persistent Threats detecting targeted attacks in cloud computing	Conference	2014	(Vance, 2014)	Multi-Confirmations and DNS Graph Mining for Malicious Domain Detection	Conference	2019	(Tran et al., 2019)
Semi-supervised classification system for the detection of advanced persistent threats	Journal	2016	(Barceló-Rico et al., 2016)	Weakly Supervised Deep Learning for the Detection of Domain Generation Algorithms	Journal	2019	(Yu et al., 2019)
DNS Tunneling Detection Techniques – Classification, and Theoretical Comparison in Case of a Real APT Campaign	Conference	2017	(Nuojua et al., 2017)	Thwarting C2 Communication of DGA-Based Malware using Process-level DNS Traffic Tracking	Conference	2019	(Menon, 2019)
Advanced persistent threat detection based on network traffic noise pattern and analysis	Journal	2016	(Ng and Bakhtiarib, 2016)	A Machine Learning Framework for Domain Generation Algorithm-Based Malware Detection	Journal	2019	(Y. Li et al., 2019)
Defense against advanced persistent threats with expert system for internet of things	Conference	2017	(Hu et al., 2017)	Anomaly Detection by Monitoring Unintended DNS Traffic on Wireless Network	Conference	2019	(Jin et al., 2019)
Polymorphic Malicious JavaScript Code Detection for APT Attack Defense	Journal	2015	(J. Choi et al., 2015)	Machine Learning-based Detection of C&C Channels with a Focus on the Locked Shields Cyber Defense Exercise	Conference	2019	(Känzig et al., 2019)
Beyond blacklisting: Cyber defense in the era of advanced persistent threats	Journal	2014	(Beuhring and Salous, 2014)	Integrating an Attention Mechanism and Deep Neural Network for Detection of DGA Domain Names	Conference	2019	(Ren et al., 2019)
Unsupervised Detection of APT C&C Channels using Web Request Graphs	Conference	2017	(Lamprakis et al., 2017)	CCGA: Clustering and Capturing Group Activities for DGA-Based Botnets Detection	Conference	2019	(Liu et al., 2019)

(continued on next page)

Table 8
(continued)

Title	Type	Year	Ref.	Title	Type	Year	Ref.
Method for Behavior-Prediction of APT attack based on Dynamic Bayesian Game	Conference	2016	(Haopu, 2016)	Abnormal Behavior Detection to Identify Infected Systems Using the APChain Algorithm and Behavioral Profiling	Journal	2018	(Seo and Lee, 2018)
The APT Detection Method in SDN	Conference	2017	(Shan-Shan and Ya-Bin, 2017)	Characterizing Network Behavior Features Using a Cyber-Security Ontology	Conference	2016	(Ben-Asher et al., 2016)
A Novel Deep Learning Stack for APT Detection	Journal	2019	(Bodström and Hämäläinen, 2019)	Advanced malicious beaconing detection through AI	Journal	2020	(Borchani, 2020)
Proposed approach for targeted attacks detection	Conference	2016	(Ghafir and Prenosil, 2016)	Detection and Classification of Different Botnet C&C Channels	Conference	2011	(Fedynyshyn et al., 2011)
N-victims: An approach to determine n-victims for apt investigations	Conference	2012	(Liu et al., 2012)	Botnet Traffic Detection Techniques by C&C Session Classification Using SVM	Conference	2007	(Kondo and Sato, 2007)
Evidence-Based Detection of Advanced Persistent Threats	Journal	2018	(Tecuci et al., 2018)	A New Hybrid Approach for C&C Channel Detection	Conference	2019	(Jiang et al., 2019)
Dealing with advanced persistent threats in smart grid ICT networks	Conference	2014	(F. Skopik et al., 2014)	C&C Session Detection using Random Forest	Article	2017	(Lu et al., 2017)
Preventing advanced persistent threats in complex control networks	Conference	2017	(Rubio et al., 2017)				
Identifying APT malware domain based on mobile DNS logging	Journal	2017	(Niu et al., 2017)				
A Study on Efficient Log Visualization Using D3 Component against APT: How to Visualize Security Logs Efficiently?	Conference	2016	(Lee et al., 2016)				
A baseline for unsupervised advanced persistent threat detection in system-level provenance	Journal	2020	(Berrada et al., 2020)				

Table 9
Selected Vendor Projects.

ID	Name	License	Updated year	ID	Name	License	Updated year
P1	Deterrent	OS	2017	P17	Imperva	Comm.	2020
P2	Aptdetector	OS	2019	P18	ArcSight	Comm.	2019
P3	XCOM	OS	2015	P19	Splunk	Comm.	2021
P4	ADAPT	OS	2019	P20	Barracuda	Comm.	2020
P5	APThreatDetectionSys	OS	2016	P21	Cisco	Comm.	2021
P6	aptdetector-go	OS	2016	P22	Fidelis	Comm.	2021
P7	ADAPTS	OS	2018	P23	FireEye	Comm.	2020
P8	MalwareModels	OS	2019	P24	Forcepoint	OS	2021
P9	Judge-Query-and-Executable	OS	2019	P25	Sophos	Comm.	2021
P10	ludumdare32	OS	2015	P26	Kaspersky	Comm.	2021
P11	THOR APT Scanner	Comm.	2021	P27	McAfee	Comm.	2021
P12	Kaspersky Security Operation Center	Comm.	2021	P28	Red Canary	OS	2020
P13	Symantec Endpoint APT protection	Comm.	2020	P29	Symantec	Comm.	2021
P14	IBM QRadar SIEM	Comm.	2019	P30	Trend Micro	Comm.	2020
P15	RSA NetWitness Platform	Comm.	2018	P31	Webroot	Comm.	2021
P16	SolarWinds	Comm.	2020				

Table 10

QAR Scores.

QAR	QAR1	QAR2	QAR3	QAR4	QAR5	QAR6	QAR7	QAR8	QAR9	QAR10	Total
(Bryant and Saiedian, 2017)	1	1	0.75	0.5	0.75	0	0.25	0.5	0.25	0.5	5.5
(Huang and Zhu, 2019)	1	0.5	1	0.75	0.75	0	0.5	0	0	0.75	5.25
(Chandran et al., 2015)	1	0.75	1	0.75	0.5	0	0.5	1	1	0.5	7
(M. Marchetti et al., 2016)	1	0.5	1	0.75	1	0	0.75	0	1	0.75	6.75
(Schindler, 2018)	1	1	1	1	1	0	1	0	1	1	8
(Hu et al., 2016)	1	1	1	1	1	0	1	0.5	1	1	8.5
(M. Marchetti et al., 2016)	1	0.5	1	1	1	0.25	1	0	0.75	1	7.5
(Li et al., 2018)	1	0.75	1	1	0.75	0.75	0.75	0	0	1	7
(Siddiqui et al., 2016)	1	1	1	1	1	0	0.75	1	1	1	8.75
(I. Ghafir et al., 2018)	1	1	1	1	0.75	0.5	0.5	1	1	1	8.75
(Lv et al., 2019)	1	1	1	1	1	0	1	0	1	1	8
(Ghafir et al., 2019)	1	1	1	1	1	1	1	0.5	1	1	9.5
(Yan et al., 2020)	1	1	1	1	1	1	1	1	1	1	10
(Baksi and Upadhyaya, 2016)	1	1	1	1	0.75	0.75	0.25	0	0	1	6.75
(Atapour et al., 2018)	1	1	1	1	1	0	0.5	0.25	1	1	7.75
(Zimba et al., 2020)	1	1	1	1	1	0.5	1	1	1	1	9.5
(Zhu and Rass, 2018)	1	1	1	1	1	0.5	0.75	0	0	0.75	7
(Cao, 2019)	1	1	1	1	1	0.75	1	0	1	1	8.75
(Virvilis and Gritzalis, 2013)	1	0.75	0.75	0.75	0.5	0.5	0	0	0	0.75	5
(Kim and Park, 2014)	1	0.75	1	0.75	0	1	1	0	1	1	7.5
(Friedberg et al., 2015)	1	1	1	1	1	1	1	0.75	1	1	9.75
(Lu et al., 2016)	1	0.5	1	1	1	1	1	1	1	1	9.5
(Giura and Wang, 2012)	1	1	0.75	1	0.75	0.75	0	1	1	1	8.25
(Lu et al., 2019)	1	1	1	1	0.75	0.25	1	1	1	1	9
(Shenwen et al., 2015)	1	1	0.75	0.75	1	0.5	0.5	0	1	1	7.5
(Mirza et al., 2014)	1	0.5	1	1	0.25	0.25	0.75	0	0	1	5.75
(Milajerdi et al., 2019)	1	1	1	1	1	1	1	0.5	1	1	9.5
(Moon et al., 2017)	1	0.75	1	1	1	1	1	1	1	1	9.75
(Wang et al., 2016)	1	1	1	1	1	1	1	0	1	1	9
(Manzoor et al., 2016)	1	1	1	1	1	0	1	1	1	1	9
(Saud and Islam, 2015)	1	0.75	1	1	1	0.25	0.75	0	0	1	6.75
(Ge et al., 2016)	1	0.5	1	1	1	0.25	1	0	1	1	7.75
(Ghafir et al., 2017)	1	0.75	1	1	0.5	0.25	1	0	1	1	7.5
(Sharma et al., 2017)	1	1	1	1	1	1	1	1	1	1	10
(Bhatt et al., 2014)	1	1	1	1	0.5	0.25	0.75	0	0	1	6.5
(Sexton et al., 2016)	1	1	1	1	1	1	1	0.25	1	1	9.25
(Brogi and Tong, 2016)	1	1	1	1	1	0.5	1	0	1	1	8.5
(J. Choi et al., 2015)	1	1	1	1	0.25	0.25	0.5	0	1	1	7
(Su et al., 2022)	1	0.75	0.75	0.75	1	0	1	0	1	0	6.25
(Zhao et al., 2015)	1	1	1	1	1	1	1	0.75	1	1	9.75
(Xiao et al., 2018)	1	1	1	1	1	1	1	0	0	1	8
(de Vries et al., 2012)	1	1	1	1	1	1	0.5	0	0	1	7.5
(Wang et al., 2014)	1	1	0.75	0.75	1	1	1	0	1	1	8.5
(Vance, 2014)	1	0.75	1	1	1	1	1	0.25	1	1	9
(Barceló-Rico et al., 2016)	1	1	0.75	0.75	1	1	1	1	1	1	9.5
(Nuojua et al., 2017)	1	0.75	1	1	0.5	0.5	1	0.75	1	1	8.5
(Ng and Bakhtiarib, 2016)	1	1	1	1	0	0.25	1	0	1	1	7.25
(Hu et al., 2017)	1	1	1	1	0.5	0.5	0.75	0	0	1	6.75
(Choi et al., 2015)	1	0.5	1	1	1	1	1	0.75	1	1	9.25
(Beuhring and Salous, 2014)	1	0.5	0.5	0.5	0.5	1	0.25	0.25	0	0.5	5
(Lamprakis et al., 2017)	1	1	1	1	1	1	1	0.5	1	1	9.5
(Haopu, 2016)	1	0.75	1	1	1	1	1	0	0.5	1	8.25
(Shan-Shan and Ya-Bin, 2017)	1	0.5	1	1	0.5	1	1	0.5	0.25	1	7.75
(Bodström and Hämäläinen, 2019)	1	0.75	1	1	1	1	1	0	0	1	7.75
(Ghafir and Prenosil, 2016)	1	0.75	1	1	1	0.75	1	0	1	1	8.5
(Liu et al., 2012)	1	1	0.75	1	1	1	0.75	0.25	0.75	1	8.5
(Tecuci et al., 2018)	1	1	1	1	1	0.75	1	0	0	1	7.75
(Skopik et al., 2014)	1	0.75	1	1	0.5	0.25	0.75	0	0	1	6.25
(Rubio et al., 2017)	1	0.25	1	1	0.25	0.25	0.75	0.5	0	1	6
(Niu et al., 2017)	1	1	1	1	1	0	1	1	1	1	9
(Lee et al., 2016)	1	1	1	1	1	0	0.75	0	1	1	7.75
(Berrada et al., 2020)	1	1	1	1	0.75	0.25	0.75	1	0.75	1	8.5
(Vert et al., 2018)	1	1	1	1	1	0.25	0.5	0.5	0.25	0.75	7.25
(Y. Li et al., 2019)	1	1	0.75	0.75	0.75	0	0.5	0.25	0	0.75	5.75
(Shi et al., 2018)	1	1	1	1	1	0	1	0.25	1	1	8.25
(Cho and Nam, 2019)	1	0.75	1	1	1	1	1	0.5	1	1	9.25
(Sengupta et al., 2019)	1	1	0.75	0.75	0.25	0.25	0.5	0.25	0	0.75	5.5
(Kim et al., 2018)	1	0.5	0.5	0.5	0.5	0.75	0	0	0.75	0.5	5
(Cui et al., 2019)	1	0.75	1	1	1	1	1	0.25	1	1	9
(Debatty et al., 2018)	1	1	1	1	0.5	0.5	1	0	1	1	8

(continued on next page)

Table 10 (continued)

QAR	QAR1	QAR2	QAR3	QAR4	QAR5	QAR6	QAR7	QAR8	QAR9	QAR10	Total
(Liu et al., 2013)	1	1	1	1	1	0	1	0	1	1	8
(Chu et al., 2019)	1	1	1	1	1	0	1	1	1	1	9
(Bencsáth et al., 2012)	1	0.5	0.5	0.5	1	1	1	0	0.25	0.5	6.25
(Bodström and Hämäläinen, 2018)	1	1	1	0.75	1	1	0.75	0	0	1	7.5
(Shan-Shan and Ya-Bin, 2018)	1	1	1	1	0.5	0.5	1	0.25	1	1	8.25
(Sigholm and Bang, 2013)	1	1	1	1	0.75	0.5	1	0.25	1	1	8.5
(Jia et al., 2015)	1	1	0.75	0.5	0.25	0.25	0.25	0.25	0	0.75	5
(Laurenza et al., 2017)	1	1	1	1	0.75	0.75	1	0.5	1	1	9
(Moothedath et al., 2020)	1	1	1	1	0.25	0.25	0.75	0.25	0.75	0.75	7
(Rass et al., 2017)	1	1	1	1	1	0	0.25	0.75	0.5	0.75	7.25
(Hu et al., 2015)	1	1	1	0.75	0.5	0	0.25	0	0	0.75	5.25
(Xiong et al., 2020)	1	1	1	1	1	0	1	0.25	1	1	8.25
(Maccari et al., 2018)	1	1	1	0.75	0.75	0.75	0.25	0	1	1	7.5
(Lee et al., 2017)	1	0.5	0.75	0.75	1	1	1	0	0.75	1	7.75
(Ghafir et al., 2018)	1	1	1	1	1	0.75	1	0	1	1	8.75
(Balduzzi et al., 2013)	1	1	1	1	1	1	1	0	1	1	9
(Chandra et al., 2016)	1	0.75	0.75	0.75	0.75	0.75	0.5	1	1	0.75	8
(Yan et al., 2019)	1	1	1	1	1	0.75	1	1	1	1	9.75
(Abdullayeva, 2021)	1	1	1	1	1	0.25	1	1	1	1	9.25
(Najafi et al., 2021)	1	1	1	0.75	0.25	0.75	0.25	0.75	0.25	1	7
(Stoleriu et al., 2021)	1	1	1	0.25	0.25	0.25	1	0.5	0	1	6.25
(Niu et al., 2021)	1	1	1	1	0.25	0.25	1	1	0.5	1	8
(Dijk, 2021)	1	1	1	1	0.75	0.75	1	1	1	1	9.5
(Oprea et al., 2018)	1	1	1	1	0.75	1	1	0.75	1	1	9.5
(Apruzzese et al., 2017)	1	1	1	0.75	0.75	1	1	1	1	1	9.5
(Richer, 2017)	1	1	0.5	0.75	0.5	0.25	0.75	0	1	1	6.75
(Ren et al., 2020)	1	0.75	1	1	1	1	1	1	0.75	1	9.5
(Highnam et al., 2021)	1	0.75	1	1	1	1	1	1	0.5	1	9.25
(Sivaguru et al., 2020)	1	1	1	1	1	1	1	1	1	1	10
(Chen et al., 2020)	1	1	1	1	0.75	0.5	0.75	1	1	1	9
(Zhou et al., 2019)	1	0.75	1	1	1	0.5	1	1	0.5	1	8.75
(Tong et al., 2019)	1	1	1	0.75	0.5	0.5	1	0.75	0.25	1	7.75
(Yan et al., 2019)	1	1	1	1	1	1	1	1	1	1	10
(Vishvakarma et al., 2020)	1	1	0.75	0.75	1	1	1	0	0.75	1	8.25
(Vinayakumar et al., 2019)	1	1	1	1	1	1	1	1	1	1	10
(Tran et al., 2019)	1	1	1	1	1	1	1	0	1	1	9
(Yu et al., 2019)	1	1	1	1	0.75	0.75	1	1	1	1	9.5
(Menon, 2019)	1	1	0.75	0.75	0.25	0	0.5	0	0.75	0.75	5.75
(Li et al., 2019)	1	1	1	1	0.75	0	1	0.75	1	1	8.5
(Jin et al., 2019)	1	1	1	1	0.75	0.75	0.75	0.75	0.25	1	8.25
(Känzig et al., 2019)	1	0.75	1	0.75	0.5	0.35	0.5	0.5	0.25	1	6.6
(Ren et al., 2019)	1	1	0.75	1	0.75	0.5	0.75	1	1	1	8.75
(Liu et al., 2019)	1	1	1	1	1	0.5	1	0.75	1	1	9.25
(Seo and Lee, 2018)	1	0.75	1	1	1	0	1	0.75	1	1	8.5
(Ben-Asher et al., 2016)	1	0.5	1	1	0.25	0.25	0.25	0.25	0.5	1	6
(Borchani, 2020)	1	0.75	0.75	1	0.25	0.25	0.5	0.25	0.5	1	6.25
(Fedynyshyn et al., 2011)	1	1	1	1	1	0.75	1	0.5	1	1	9.25
(Kondo and Sato, 2007)	1	0.75	1	1	0.5	0.5	0.75	1	0.75	1	8.25
(Jiang et al., 2019)	1	1	1	1	0.75	0	1	0.25	1	1	8
(Lu et al., 2017)	1	1	1	1	0.5	0.5	1	0.25	1	1	8.25

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:[10.1016/j.cose.2022.102875](https://doi.org/10.1016/j.cose.2022.102875).

References

- Li, M., Huang, W., Wang, Y., Fan, W., Li, J., 2016. The study of APT attack stage model. In: Proceedings of the IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), pp. 1–5. doi:[10.1109/ICIS.2016.7550947](https://doi.org/10.1109/ICIS.2016.7550947).
- Vukalović, J., Delija, D., 2015. Advanced Persistent Threats - detection and defense. In: Proceedings of the 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 1324–1330. doi:[10.1109/MIPRO.2015.7160480](https://doi.org/10.1109/MIPRO.2015.7160480).
- D. McWhorter, "Mandiant exposes APT1—one of China's cyber espionage units & releases 3,000 indicators," Mandiant Febr., vol. 18, 2013.
- Stojanović, B., Hofer-Schmitz, K., Kleb, U., 2020. APT datasets and attack modeling for automated detection methods: a review. *Comput. Secur.* 92, 101734. doi:[10.1016/j.cose.2020.101734](https://doi.org/10.1016/j.cose.2020.101734).
- Brewer, R., 2014. Advanced persistent threats: minimising the damage. *Netw. Secur.* 2014 (4), 5–9. doi:[10.1016/S1353-4858\(14\)70040-6](https://doi.org/10.1016/S1353-4858(14)70040-6).
- Ussath, M., Jaeger, D., Cheng, F., Meinel, C., 2016. Advanced persistent threats: behind the scenes. In: Proceedings of the Annual Conference on Information Science and Systems (CISS), pp. 181–186. doi:[10.1109/CISS.2016.7460498](https://doi.org/10.1109/CISS.2016.7460498).
- Messaoud, B.I.D., Guennoun, K., Wahbi, M., Sadik, M., 2016. Advanced Persistent Threat: new analysis driven by life cycle phases and their challenges. In: Proceedings of the International Conference on Advanced Communication Systems and Information Security (ACOSIS), pp. 1–6. doi:[10.1109/ACOSIS.2016.7843932](https://doi.org/10.1109/ACOSIS.2016.7843932).
- Virvilis, N., Gritzalis, D., Apostolopoulos, T., 2013. Trusted computing vs. advanced persistent threats: can a defender win this game? In: Proceedings of the IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing, pp. 396–403. doi:[10.1109/UIC-ATC.2013.80](https://doi.org/10.1109/UIC-ATC.2013.80).
- Chen, P., Desmet, L., Huygens, C., 2014. A study on advanced persistent threats. *Commun. Multimed. Secur.* 63–72.
- Alshamrani, A., Myneni, S., Chowdhary, A., Huang, D., 2019. A Survey on advanced persistent threats: techniques, solutions, challenges, and research opportunities. *IEEE Commun. Surv. Tutorials* 21 (2), 1851–1877. doi:[10.1109/COMST.2019.2891891](https://doi.org/10.1109/COMST.2019.2891891).
- Quintero-Bonilla, S., del Rey, A., 2020a. A new proposal on the advanced persistent threat: a survey. *Appl. Sci.* 10 (11). doi:[10.3390/app10113874](https://doi.org/10.3390/app10113874).
- Rajalakshmi, E., Asik Ibrahim, N., Subramaniaswamy, V., 2019. A survey of machine learning techniques used to combat against the advanced persistent threat. *Appl. Tech. Inf. Secur.* 159–172.
- Quintero-Bonilla, S., del Rey, A.M., 2020b. Proposed models for advanced persistent threat detection: a review. In: Distributed Computing and Artificial Intelligence, 16th International Conference, Special Sessions, pp. 141–148.
- Singh, S., Sharma, P.K., Moon, S.Y., Moon, D., Park, J.H., 2019. A comprehensive study on APT attacks and countermeasures for future networks and communications: challenges and solutions. *J. Supercomput.* 75 (8), 4543–4574. doi:[10.1007/s11227-016-1850-4](https://doi.org/10.1007/s11227-016-1850-4).
- Lemay, A., Calvet, J., Menet, F., Fernandez, J.M., 2018. Survey of publicly available reports on advanced persistent threat actors. *Comput. Secur.* 72, 26–59. doi:[10.1016/j.cose.2017.08.005](https://doi.org/10.1016/j.cose.2017.08.005).
- Nissim, N., Cohen, A., Glezer, C., Elovici, Y., 2015. Detection of malicious PDF files and directions for enhancements: a state-of-the-art survey. *Comput. Secur.* 48, 246–266. doi:[10.1016/j.cose.2014.10.014](https://doi.org/10.1016/j.cose.2014.10.014).
- Luh, R., Marschalek, S., Kaiser, M., Janicke, H., Schrittwieser, S., 2017. Semantics-aware detection of targeted attacks: a survey. *J. Comput. Virol. Hacking Tech.* 13 (1), 47–85. doi:[10.1007/s11416-016-0273-3](https://doi.org/10.1007/s11416-016-0273-3).
- Ahmad, A., Webb, J., Desouza, K.C., Boorman, J., 2019. Strategically-motivated advanced persistent threat: definition, process, tactics and a disinformation model of counterattack. *Comput. Secur.* 86, 402–418. doi:[10.1016/j.cose.2019.07.001](https://doi.org/10.1016/j.cose.2019.07.001).
- S. Keele and others, "Guidelines for performing systematic literature reviews in software engineering," 2007.
- Zhao, G., Xu, K., Xu, L., Wu, B., 2015. Detecting APT Malware infections based on malicious DNS and traffic analysis. *IEEE Access* 3, 1132–1142. doi:[10.1109/ACCESS.2015.2458581](https://doi.org/10.1109/ACCESS.2015.2458581).
- Lu, J., Chen, K., Zhuo, Z., Zhang, X., 2019. A temporal correlation and traffic analysis approach for APT attacks detection. *Cluster Comput.* 22 (3), 7347–7358. doi:[10.1007/s10586-017-1256-y](https://doi.org/10.1007/s10586-017-1256-y).
- Cho, D.X., Nam, H.H., 2019. A method of monitoring and detecting APT attacks based on unknown domains. *Procedia Comput. Sci.* 150, 316–323. doi:[10.1016/j.procs.2019.02.058](https://doi.org/10.1016/j.procs.2019.02.058).
- E. Manzoor, S. Milajerdi, and L. Akoglu, "Fast memory-efficient anomaly detection in streaming heterogeneous graphs," 2016, pp. 1035–1044, doi:[10.1145/2939672.2939783](https://doi.org/10.1145/2939672.2939783).
- Choi, J., Choi, C., You, I., Kim, P., 2015a. Polymorphic Malicious JavaScript Code Detection for APT Attack Defence. *J. Univers. Comput. Sci.* 21, 369–383. doi:[10.3217/jucs-021-03-0369](https://doi.org/10.3217/jucs-021-03-0369).
- Zimba, A., Chen, H., Wang, Z., Chishimba, M., 2020. Modeling and detection of the multi-stages of Advanced Persistent Threats attacks based on semi-supervised learning and complex networks characteristics. *Futur. Gener. Comput. Syst.* 106, 501–517. doi:[10.1016/j.future.2020.01.032](https://doi.org/10.1016/j.future.2020.01.032).
- Rass, S., König, S., Schauer, S., 2017. Defending against advanced persistent threats using game-theory. *PLoS One* 12 (1), e0168675.
- Sengupta, S., Chowdhary, A., Huang, D., Kambhampati, S., 2019. General Sum Markov games for strategic detection of advanced persistent threats using moving target defense in cloud networks. *Decis. Game Theory Secur.* 492–512.
- Haopo, Y., 2016. Method for behavior-prediction of APT attack based on dynamic Bayesian game. In: Proceedings of the IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), pp. 177–182. doi:[10.1109/ICCCBDA.2016.7529554](https://doi.org/10.1109/ICCCBDA.2016.7529554).
- Huang, L., Zhu, Q., 2019. Adaptive strategic cyber defense for advanced persistent threats in critical infrastructure networks. *SIGMETRICS Perform. Eval. Rev.* 46 (2), 52–56. doi:[10.1145/3305218.3305239](https://doi.org/10.1145/3305218.3305239).
- Virvilis, N., Gritzalis, D., 2013. The big four - what we did wrong in advanced persistent threat detection? In: Proceedings of the International Conference on Availability, Reliability and Security, pp. 248–254. doi:[10.1109/ARES.2013.32](https://doi.org/10.1109/ARES.2013.32).
- Ghafir, I., et al., 2019. Hidden MMarkov models and alert correlations for the prediction of advanced persistent threats. *IEEE Access* 7, 99508–99520. doi:[10.1109/ACCESS.2019.2930200](https://doi.org/10.1109/ACCESS.2019.2930200).
- de Vries, J., Hoogstraaten, H., van den Berg, J., Daskapan, S., 2012. Systems for detecting advanced persistent threats: a development roadmap using intelligent data analysis. In: Proceedings of the International Conference on Cyber Security, pp. 54–61. doi:[10.1109/CyberSecurity.2012.14](https://doi.org/10.1109/CyberSecurity.2012.14).
- Liu, S.-T., Chen, Y.-M., Hung, H.-C., 2012. N-victims: an approach to determine N-victims for APT investigations. *Inf. Secur. Appl.* 226–240.
- Bencsáth, B., Pék, G., Buttyán, L., Félégyházi, M., 2012. Duqu: analysis, detection, and lessons learned. *ACM Eur. Workshop Syst. Secur. (EuroSec)* 2012.
- Liu, S.T., Chen, Y.M., Lin, S.J., 2013. A novel search engine to uncover potential victims for APT investigations. *Netw. Parallel Comput.* 405–416.
- Sigholm, J., Bang, M., 2013. Towards offensive cyber counterintelligence: adopting a target-centric view on advanced persistent threats. In: Proceedings of the European Intelligence and Security Informatics Conference, pp. 166–171. doi:[10.1109/EISIC.2013.37](https://doi.org/10.1109/EISIC.2013.37).
- Najafi, P., Cheng, F., Meinel, C., 2021. SIEMA: bringing advanced analytics to legacy security information and event management. *Secur. Privacy Commun. Netw.* 25–43.
- Bryant, B.D., Saiedian, H., 2017. A novel kill-chain framework for remote security log analysis with SIEM software. *Comput. Secur.* 67, 198–210. doi:[10.1016/j.cose.2017.03.003](https://doi.org/10.1016/j.cose.2017.03.003).
- Atapour, C., Agrafiotis, I., Creese, S., 2018. Modeling Advanced Persistent Threats to enhance anomaly detection techniques. *J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl.* 9 (4), 71–102.
- Bodström, T., Hämäläinen, T., 2018. A Novel Method for Detecting APT Attacks by Using OODA Loop and Black Swan Theory. *Comput. Data Soc. Netw.* 498–509.
- Marchetti, M., Pierazzi, F., Colajanni, M., Guido, A., 2016a. Analysis of high volumes of network traffic for Advanced Persistent Threat detection. *Comput. Netw.* 109, 127–141. doi:[10.1016/j.comnet.2016.05.018](https://doi.org/10.1016/j.comnet.2016.05.018).
- Marchetti, M., Pierazzi, F., Guido, A., Colajanni, M., 2016b. Countering Advanced Persistent Threats through security intelligence and big data analytics. In: Proceedings of the 8th International Conference on Cyber Conflict (CyCon), pp. 243–261. doi:[10.1109/CYCON.2016.7529438](https://doi.org/10.1109/CYCON.2016.7529438).
- Yan, G., Li, Q., Guo, D., Meng, X., 2020. Discovering suspicious APT behaviors by analyzing DNS activities. *Sensors* 20 (3). doi:[10.3390/s20030731](https://doi.org/10.3390/s20030731).
- Friedberg, I., Skopik, F., Settanni, G., Fiedler, R., 2015. Combating advanced persistent threats: from network event correlation to incident detection. *Comput. Secur.* 48, 35–57. doi:[10.1016/j.cose.2014.09.006](https://doi.org/10.1016/j.cose.2014.09.006).
- Choi, J., Choi, C., Lynn, H.M., Kim, P., 2015b. Ontology based APT attack behavior analysis in cloud computing. In: Proceedings of the 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA), pp. 375–379. doi:[10.1109/BWCCA.2015.69](https://doi.org/10.1109/BWCCA.2015.69).
- Su, Y., Li, M., Tang, C., Shen, R., 2022. A framework of APT detection based on dynamic analysis. In: Proceedings of the 2015 4th National Conference on Electrical, Electronics and Computer Engineering, pp. 1047–1053. doi:[10.2991/nceee-15.2016.187](https://doi.org/10.2991/nceee-15.2016.187).
- Wang, Y., Wang, Y., Liu, J., Huang, Z., 2014. A network gene-based framework for detecting advanced persistent threats. In: Proceedings of the Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, pp. 97–102. doi:[10.1109/3PGCIC.2014.41](https://doi.org/10.1109/3PGCIC.2014.41).
- Vance, A., 2014. Flow based analysis of Advanced Persistent Threats detecting targeted attacks in cloud computing. In: Proceedings of the First International Scientific-Practical Conference Problems of Infocommunications Science and Technology, pp. 173–176. doi:[10.1109/INFOCOMMST.2014.6992342](https://doi.org/10.1109/INFOCOMMST.2014.6992342).
- Nuojua, V., David, G., Hämäläinen, T., 2017. DNS tunneling detection techniques – classification, and theoretical comparison in case of a real APT campaign. *Internet Things Smart Space Next Gener. Netw. Syst.* 280–291.
- Ng, S., Bakhtiar, M., 2016. Advanced persistent threat detection based on network traffic noise pattern and analysis. *J. Adv. Res. Comput. Appl.* 21, 1–18.
- Ghafir, I., Prenosil, V., 2016. Proposed approach for targeted attacks detection. *Adv. Comput. Commun. Eng. Technol.* 73–80.
- Jia, B., Lin, Z., Ma, Y., 2015. Advanced Persistent Threat Detection method research based on relevant algorithms to artificial immune system. *Trustworthy Comput. Serv.* 221–228.
- Stoleriu, R., Puncioiu, A., Bica, I., 2021. Cyber attacks detection using open source ELK stack. In: Proceedings of the 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), pp. 1–6. doi:[10.1109/ECAI52376.2021.9515120](https://doi.org/10.1109/ECAI52376.2021.9515120).

- Giura, P., Wang, W., 2012. A context-based detection framework for advanced persistent threats. In: Proceedings of the International Conference on Cyber Security, pp. 69–74. doi:10.1109/CyberSecurity.2012.16.
- Mirza, N.A.S., Abbas, H., Khan, F.A., Al Muhtadi, J., 2014. Anticipating Advanced Persistent Threat (APT) countermeasures using collaborative security mechanisms. In: 2014 International Symposium on Biometrics and Security Technologies (IS-BAST), pp. 129–132. doi:10.1109/ISBAST.2014.7013108.
- Sharma, P.K., Moon, S.Y., Moon, D., Park, J.H., 2017. DFA-AD: a distributed framework architecture for the detection of advanced persistent threats. Clust. Comput. 20 (1), 597–609. doi:10.1007/s10586-016-0716-0.
- Bhatt, P., Yano, E.T., Gustavsson, P., 2014. Towards a framework to detect multi-stage advanced persistent threats attacks. In: Proceedings of the IEEE 8th International Symposium on Service Oriented System Engineering, pp. 390–395. doi:10.1109/SOSE.2014.53.
- Brogi, G., Tong, V.V.T., 2016. TerminAPTor: highlighting advanced persistent threats through information flow tracking. In: Proceedings of the 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS), pp. 1–5. doi:10.1109/NTMS.2016.7792480.
- Shan-Shan, J., Ya-Bin, X., 2018. The APT detection method based on attack tree for SDN. In: Proceedings of the 2nd International Conference on Cryptography, Security and Privacy, pp. 116–121. doi:10.1145/3199478.3199481.
- Maccari, M., Polzonetti, A., Sagratella, M., 2019. Detection: definition of new model to reveal advanced persistent threat. In: *Proceedings of the Future Technologies Conference (FTC) 2018*, pp. 305–323.
- T. Schindler, “Anomaly detection in log data using graph databases and machine learning to defend advanced persistent threats,” 2018, doi: 10.18420/in2017_241.
- Milajerdi, S.M., Gjomemo, R., Eshete, B., Sekar, R., Venkatakrishnan, V.N., 2019. HOLMES: real-time APT Detection through Correlation of Suspicious Information Flows. In: Proceedings of the IEEE Symposium on Security and Privacy (SP), pp. 1137–1152. doi:10.1109/SP.2019.00026.
- Lamprakis, P., Dargenio, R., Gugelmann, D., Lenders, V., Happe, M., Vanbever, L., 2017. Unsupervised detection of APT C&C channels using web request graphs. *Detect.Intrus. Malware Vulner. Assess.* 366–387.
- Rubio, J.E., Alcaraz, C., Lopez, J., 2017. Preventing advanced persistent threats in complex control networks. *Comput. Secur. ESORICS 2017* 402–418.
- Debatty, T., Mees, W., Gilon, T., 2018. Graph-based APT detection. In: Proceedings of the International Conference on Military Communications and Information Systems (ICMCIS), pp. 1–8. doi:10.1109/ICMCIS.2018.8398708.
- Do Xuan, C., Huang, D.T., 2022. A new approach for APT malware detection based on deep graph network for endpoint systems. *Appl. Intell.* doi:10.1007/s10489-021-03138-z.
- Saud, Z., Islam, M.H., 2015. Towards proactive detection of advanced persistent threat (APT) attacks using honeypots. In: Proceedings of the 8th International Conference on Security of Information and Networks, pp. 154–157. doi:10.1145/2799979.2800042.
- Lee, J., Jeon, J., Lee, C., Lee, J., Cho, J., Lee, K., 2016. A Study on Efficient Log Visualization Using D3 Component against APT: how to Visualize Security Logs Efficiently? In: 2016 International Conference on Platform Technology and Service (PlatCon), pp. 1–6. doi:10.1109/PlatCon.2016.7456778.
- Beuhring, A., Salous, K., 2014. Beyond blacklisting: cyberdefense in the era of advanced persistent threats. *IEEE Secur. Priv.* 12 (5), 90–93. doi:10.1109/MSP.2014.86.
- Skopik, F., Friedberg, I., Fiedler, R., 2014a. Dealing with advanced persistent threats in smart grid ICT networks. *ISGT 2014* 1–5. doi:10.1109/ISGT.2014.6816388.
- P. Cao, “On preempting advanced persistent threats using probabilistic graphical models,” arXiv Prepr. *arXiv1903.08826*, 2019.
- Kim, Y.-H., Park, W.H., 2014. A study on cyber threat prediction based on intrusion detection event for APT attack detection. *Multimed. Tools Appl.* 71 (2), 685–698. doi:10.1007/s11042-012-1275-x.
- Cui, Y., Xue, J., Wang, Y., Liu, Z., Zhang, J., 2019. Research of snort rule extension and APT detection based on APT network behavior analysis. *Trusted Comput. Inf. Secur.* 51–64.
- I. Ghafir, V. Prenosil, M. Hammoudeh, L. Han, and U. Raza, “Malicious SSL Certificate Detection: a Step Towards Advanced Persistent Threat Defence,” 2017, doi: 10.1145/3102304.3102331.
- I. Ghafir, V. Prenosil, M. Hammoudeh, F.J. Aparicio-Navarro, K. Rabie, and A. Jabban, “Disguised executable files in spear-phishing emails: detecting the point of entry in advanced persistent threat,” 2018, doi: 10.1145/3231053.3231097.
- Hu, X., et al., 2016. BAYWATCH: robust beaconing detection to identify infected hosts in large-scale enterprise networks. In: Proceedings of the 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp. 479–490. doi:10.1109/DSN.2016.50.
- Bakshi, R.P., Upadhyaya, S., 2016. Kidemonas: the Silent Guardian. *World Acad. Sci. Eng. Technol. Int. J. Comput. Electr. Autom. Control Inf. Eng. Vol10 (No4) vol. abs/1712.0*, 2017.
- Kim, G., Choi, C., Choi, J., 2018. Ontology Modeling for APT Attack Detection in an IoT-Based Power System. In: Proceedings of the 2018 Conference on Research in Adaptive and Convergent Systems, pp. 160–164. doi:10.1145/3264746.3264786.
- Chandra, J.V., Challa, N., Pasupuleti, S.K., 2016. A practical approach to E-mail spam filters to protect data from advanced persistent threat. In: Proceedings of the International Conference on Circuit, Power and Computing Technologies (ICCPCT), pp. 1–5. doi:10.1109/ICCPCT.2016.7530239.
- Wang, X., Zheng, K., Niu, X., Wu, B., Wu, C., 2016. Detection of command and control in advanced persistent threat based on independent access. In: Proceedings of the IEEE International Conference on Communications (ICC), pp. 1–6. doi:10.1109/ICC.2016.7511197.
- Li, P., Yang, X., Xiong, Q., Wen, J., Tang, Y.Y., 2018. Defending against the advanced persistent threat: an optimal control approach. *Secur. Commun. Networks* 2018, 2975376. doi:10.1155/2018/2975376.
- Lv, K., Chen, Y., Hu, C., 2019. Dynamic defense strategy against advanced persistent threat under heterogeneous networks. *Inf. Fusion* 49, 216–226. doi:10.1016/j.inffus.2019.01.001.
- Zhu, Q., Rass, S., 2018. On Multi-Phase and Multi-Stage Game-Theoretic Modeling of Advanced Persistent Threats. *IEEE Access* 6, 13958–13971. doi:10.1109/ACCESS.2018.2814481.
- Xiao, L., Xu, D., Mandayam, N.B., Poor, H.V., 2018. Attacker-Centric View of a Detection Game against Advanced Persistent Threats. *IEEE Trans. Mob. Comput.* 17 (11), 2512–2523. doi:10.1109/TMC.2018.2814052.
- Hu, Q., Lv, S., Shi, Z., Sun, L., Xiao, L., 2017. Defense Against Advanced Persistent Threats with Expert System for Internet of Things. *Wireless Algorithms, Systems, and Applications* 326–337.
- Li, Y., Zhang, T., Li, X., Li, T., 2019a. A Model of APT Attack Defense Based on Cyber Threat Detection. *Cyber Secur.* 122–135.
- Moothedath, S., et al., 2020. A Game-Theoretic Approach For Dynamic Information Flow Tracking To Detect Multistage Advanced Persistent Threats. *IEEE Trans. Automat. Contr.* 65 (12), 5248–5263. doi:10.1109/TAC.2020.2976040.
- Hu, P., Li, H., Fu, H., Cansever, D., Mohapatra, P., 2015. Dynamic defense strategy against advanced persistent threat with insiders. In: Proceedings of the IEEE Conference on Computer Communications (INFOCOM), pp. 747–755. doi:10.1109/INFOCOM.2015.7218444.
- Ge, L., Wang, L., Xu, L., 2016. An APT trojans detection method for cloud computing based on memory analysis and FCM. In: Proceedings of the 3rd International Conference on Information Science and Control Engineering (ICISCE), pp. 179–183. doi:10.1109/ICISCE.2016.48.
- Xiong, C., et al., 2020. CONAN: a practical real-time APT detection system with high accuracy and efficiency. *IEEE Trans. Depend. Secur. Comput.* 1. doi:10.1109/TDSC.2020.2971484.
- Vert, G., Claesson-Vert, A.L., Roberts, J., Bott, E., 2018. A Technology for detection of advanced persistent threat in networks and systems using a finite angular state velocity machine and vector mathematics. In: *Computer and Network Security Essentials*. Springer International Publishing, pp. 41–64. K. Daimi, Ed. Cham.
- Ren, F., Jiang, Z., Wang, X., Liu, J., 2020. A DGA domain names detection modeling method based on integrating an attention mechanism and deep neural network. *Cybersecurity* 3 (1), 4. doi:10.1186/s42400-020-00046-6.
- Highnam, K., Puzio, D., Luo, S., Jennings, N.R., 2021. Real-time detection of dictionary DGA network traffic using deep learning. *SN Comput. Sci.* 2 (2), 110. doi:10.1007/s42979-021-00507-w.
- Sivaguru, R., Peck, J., Olumofin, F., Nascimento, A., De Cock, M., 2020. Inline detection of DGA domains using side information. *IEEE Access* 8, 141910–141922. doi:10.1109/ACCESS.2020.3013494.
- Tong, M., et al., 2019. D3N: DGA detection with deep-learning through NXDomain. *Knowl. Sci. Eng. Manag.* 464–471.
- Ren, F., Jiang, Z., Liu, J., 2019. Integrating an attention mechanism and deep neural network for detection of DGA domain names. In: Proceedings of the IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), pp. 848–855. doi:10.1109/ICTAI.2019.00121.
- Dijk, A., 2021. Detection of advanced persistent threats using artificial intelligence for deep packet inspection. In: Proceedings of the IEEE International Conference on Big Data (Big Data), pp. 2092–2097. doi:10.1109/BigData52589.2021.9671464.
- Niu, W., Zhou, J., Zhao, Y., Zhang, X., Peng, Y., Huang, C., 2022. Uncovering APT malware traffic using deep learning combined with time sequence and association analysis. *Comput. Secur.* 120, 102809. doi:10.1016/j.cose.2022.102809.
- Berrada, G., et al., 2020. A baseline for unsupervised advanced persistent threat detection in system-level provenance. *Future Gener. Comput. Syst.* 108, 401–413. doi:10.1016/j.future.2020.02.015.
- Siddiqui, S., Khan, M.S., Ferens, K., Kinsner, W., 2016. Detecting advanced persistent threats using fractal dimension based machine learning classification. In: Proceedings of the 2016 ACM on International Workshop on Security And Privacy Analytics, pp. 64–69. doi:10.1145/2875475.2875484.
- Ghafir, I., et al., 2018b. Detection of advanced persistent threat using machine-learning correlation analysis. *Future Gener. Comput. Syst.* 89, 349–359. doi:10.1016/j.future.2018.06.055.
- Lu, J., Zhang, X., Junfeng, W., Lingyun, Y., 2016. APT traffic detection based on time transform. In: Proceedings of the International Conference on Intelligent Transportation, Big Data Smart City (ICITBS), pp. 9–13. doi:10.1109/ICITBS.2016.87.
- Shenwen, L., Yingbo, L., Xiongjie, D., 2015. Study and research of APT detection technology based on big data processing architecture. In: Proceedings of the IEEE 5th International Conference on Electronics Information and Emergency Communication, pp. 313–316. doi:10.1109/ICEIEC.2015.7284547.
- Barceló-Rico, F., Esparcia-Alcázar, A.I., Villalón-Huerta, A., 2016. Semi-supervised classification system for the detection of advanced persistent threats. In: *Recent Advances in Computational Intelligence in Defense and Security*. Cham: Springer International Publishing, pp. 225–248. R. Abielmona, R. Falcon, N. Zincir-Heywood, and H. A. Abbass, Eds.
- Moon, D., Im, H., Kim, I., Park, J.H., 2017. DTB-IDS: an intrusion detection system based on decision tree using behavior analysis for preventing APT attacks. *J. Supercomput.* 73 (7), 2881–2895. doi:10.1007/s11227-015-1604-8.
- Chu, W.-L., Lin, C.-J., Chang, K.-N., 2019. Detection and classification of advanced persistent threats and attacks using the support vector machine. *Appl. Sci.* 9 (21), doi:10.3390/app9214579.
- Yan, D., Zhang, H., Wang, Y., Zang, T., Xu, X., Zeng, Y., 2019a. Pontus: a linguistics-based DGA detection system. In: Proceedings of the IEEE Global Communi-

- cations Conference (GLOBECOM), pp. 1–6. doi:[10.1109/GLOBECOM38437.2019.9014040](https://doi.org/10.1109/GLOBECOM38437.2019.9014040).
- Sexton, J., Storie, C., Anderson, B., 2016. Subroutine based detection of APT malware. *J. Comput. Virol. Hacking Tech.* 12 (4), 225–233. doi:[10.1007/s11416-015-0258-7](https://doi.org/10.1007/s11416-015-0258-7).
- Shi, Y., Chen, G., Li, J., 2018. Malicious domain name detection based on extreme machine learning. *Neural Process. Lett.* 48 (3), 1347–1357. doi:[10.1007/s11063-017-9666-7](https://doi.org/10.1007/s11063-017-9666-7).
- S. Chandran, H.P., Poornachandran, P., 2015. An efficient classification model for detecting advanced persistent threat. In: Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 2001–2009. doi:[10.1109/ICACCI.2015.7275911](https://doi.org/10.1109/ICACCI.2015.7275911).
- Zhou, S., Lin, L., Yuan, J., Wang, F., Ling, Z., Cui, J., 2019. CNN-based DGA Detection with High Coverage. In: Proceedings of the IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 62–67. doi:[10.1109/ISI.2019.8823200](https://doi.org/10.1109/ISI.2019.8823200).
- Abdullayeva, F.J., 2021. Advanced Persistent Threat attack detection method in cloud computing based on autoencoder and softmax regression algorithm. *Array* 10, 100067. doi:[10.1016/j.array.2021.100067](https://doi.org/10.1016/j.array.2021.100067).
- Chen, T., Zhou, G., Liu, Z., Jing, T., 2020. A novel ensemble anomaly based approach for command and control channel detection. In: Proceedings of the 2020 4th International Conference on Cryptography, Security and Privacy, pp. 74–78. doi:[10.1145/3377644.3377652](https://doi.org/10.1145/3377644.3377652).
- Laurenza, G., Aniello, L., Lazeretti, R., Baldoni, R., 2017. Malware triage based on static features and public APT reports. *Cyber Secur. Cryptogr. Mach. Learn.* 288–305.
- Känzig, N., Meier, R., Gambazzi, L., Lenders, V., Vanbever, L., 2019. Machine learning-based detection of C channels with a focus on the locked shields cyber defense exercise. In: Proceedings of the 11th International Conference on Cyber Conflict (CyCon), 900, pp. 1–19. doi:[10.23919/CYCON.2019.8756814](https://doi.org/10.23919/CYCON.2019.8756814).
- L. Lu, Y. Feng, and K. Sakurai, “C&C session detection using random forest,” 2017, doi: [10.1145/302227.3022260](https://doi.org/10.1145/302227.3022260).
- Niu, W., et al., 2021. Malware on internet of UAVs detection combining string matching and fourier transformation. *IEEE Internet Things J.* 8 (12), 9905–9919. doi:[10.1109/IIOT.2020.3029970](https://doi.org/10.1109/IIOT.2020.3029970).
- Niu, W., Zhang, X., Yang, G., Zhu, J., Ren, Z., 2017. Identifying APT malware domain based on mobile DNS logging. *Math. Probl. Eng.* 2017, 1–9. doi:[10.1155/2017/4916953](https://doi.org/10.1155/2017/4916953).
- Bodström, T., Hämäläinen, T., 2019. A Novel Deep Learning Stack for APT Detection. *Appl. Sci.* 9 (6), doi:[10.3390/app9061055](https://doi.org/10.3390/app9061055).
- Shan-Shan, J., Ya-Bin, X., 2017. The APT detection method in SDN. In: Proceedings of the 3rd IEEE International Conference on Computer and Communications (ICCC), pp. 1240–1245. doi:[10.1109/CompComm.2017.8322741](https://doi.org/10.1109/CompComm.2017.8322741).
- Kondo, S., Sato, N., 2007. Botnet traffic detection techniques by C&C Session Classification Using SVM. In: Proceedings of the Security 2nd International Conference on Advances in Information and Computer Security, pp. 91–104.
- Yu, B., et al., 2019. Weakly supervised deep learning for the detection of domain generation algorithms. *IEEE Access* 7, 51542–51556. doi:[10.1109/ACCESS.2019.2911522](https://doi.org/10.1109/ACCESS.2019.2911522).
- Liu, Z., Yun, X., Zhang, Y., Wang, Y., 2019. CCGA: clustering and capturing group activities for DGA-based botnets detection. In: Proceedings of the 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), pp. 136–143. doi:[10.1109/TrustCom/BigDataSE.2019.00027](https://doi.org/10.1109/TrustCom/BigDataSE.2019.00027).
- Borchani, Y., 2020. Advanced malicious beaconing detection through AI. *Netw. Secur.* 2020 (3), 8–14. doi:[10.1016/S1353-4858\(20\)30030-1](https://doi.org/10.1016/S1353-4858(20)30030-1).
- Yan, G., Li, Q., Guo, D., Li, B., 2019b. AULD: large scale suspicious DNS activities detection via unsupervised learning in advanced persistent threats. *Sensors* 19 (14), doi:[10.3390/s19143180](https://doi.org/10.3390/s19143180).
- Tecuci, G., Marcu, D., Meckl, S., Boicu, M., 2018. Evidence-based detection of advanced persistent threats. *Comput. Sci. Eng.* 20 (6), 54–65. doi:[10.1109/MCSE.2018.2873854](https://doi.org/10.1109/MCSE.2018.2873854).
- Balduzzi, M., Ciangaglini, V., McArdle, R., 2013. Targeted attacks detection with SPUNge. In: Proceedings of the Eleventh Annual Conference on Privacy, Security and Trust, pp. 185–194. doi:[10.1109/PST.2013.6596053](https://doi.org/10.1109/PST.2013.6596053).
- Lee, M., Choi, J., Choi, C., Kim, P., 2017. APT attack behavior pattern mining using the FP-growth algorithm. In: Proceedings of the 14th IEEE Annual Consumer Communications Networking Conference (CCNC), pp. 1–4. doi:[10.1109/CCNC.2017.8013435](https://doi.org/10.1109/CCNC.2017.8013435).
- Xing, Y., Shu, H., Zhao, H., Li, D., Guo, L., 2021. Survey on botnet detection techniques: classification, methods, and evaluation. *Math. Probl. Eng.* 2021.
- Gaonkar, S., Dessai, N.F., Costa, J., Borkar, A., Aswale, S., Shetgaonkar, P., 2020. A Survey on Botnet Detection Techniques. In: 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), pp. 1–6. doi:[10.1109/ic-ETITE47903.2020.1d-70](https://doi.org/10.1109/ic-ETITE47903.2020.1d-70).
- Apruzzese, G., Marchetti, M., Colajanni, M., Zoccoli, G.G., Guido, A., 2017. Identifying malicious hosts involved in periodic communications. In: Proceedings of the IEEE 16th International Symposium on Network Computing and Applications (NCA), pp. 1–8. doi:[10.1109/NCA.2017.8171326](https://doi.org/10.1109/NCA.2017.8171326).
- T.J. Richer, “Entropy-based detection of botnet command and control,” 2017, doi: [10.1145/3014812.3014889](https://doi.org/10.1145/3014812.3014889).
- Vishvakarma, D.K., Bhatia, A., Riha, Z., 2020. Detection of algorithmically generated domain names in botnets. *Adv. Inf. Network. Appl.* 1279–1290.
- Jin, Y., Tomoishi, M., Yamai, N., 2019. Anomaly detection by monitoring unintended DNS traffic on wireless network. In: Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), pp. 1–6. doi:[10.1109/PACRIM47961.2019.8985052](https://doi.org/10.1109/PACRIM47961.2019.8985052).
- Seo, J., Lee, S., 2018. Abnormal behavior detection to identify infected systems using the AP chain algorithm and behavioral profiling. *Secur. Commun. Netw.* 2018.
- Ben-Asher, N., Hutchinson, S., Oltramari, A., 2016. Characterizing network behavior features using a cyber-security ontology. In: Proceedings of the MILCOM 2016 - 2016 IEEE Military Communications Conference, pp. 758–763. doi:[10.1109/MILCOM.2016.7795420](https://doi.org/10.1109/MILCOM.2016.7795420).
- Fedynyshyn, G., Chuah, M.C., Tan, G., 2011. Detection and classification of different botnet C&C channels. *Auton. Trusted Comput.* 228–242.
- Jiang, J., Yin, Q., Shi, Z., Wang, Q., Zhou, W., 2019. A new hybrid approach for C&C channel detection. In: Proceedings of the IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), pp. 583–590. doi:[10.1109/HPCC/SmartCity/DSS.2019.00090](https://doi.org/10.1109/HPCC/SmartCity/DSS.2019.00090).
- Vinayakumar, R., Soman, K.P., Poornachandran, P., Akarsh, S., Elhoseny, M., Hassanien, A.E., Elhoseny, M., 2019. Improved DGA domain names detection and categorization using deep learning architectures with classical machine learning algorithms. In: *Cybersecurity and Secure Information Systems: challenges and Solutions in Smart Environments*. Cham: Springer International Publishing, pp. 161–192.
- Oprea, A., Li, Z., Norris, R., Bowers, K., 2018. MADE: security analytics for enterprise threat detection. In: Proceedings of the 34th Annual Computer Security Applications Conference, pp. 124–136. doi:[10.1145/3274694.3274710](https://doi.org/10.1145/3274694.3274710).
- Li, Y., Xiong, K., Chin, T., Hu, C., 2019b. A machine learning framework for domain generation algorithm-based malware detection. *IEEE Access* 7, 32765–32782. doi:[10.1109/ACCESS.2019.2891588](https://doi.org/10.1109/ACCESS.2019.2891588).
- Tran, H., Dang, C., Nguyen, H., Vo, P., Vu, T., 2019. Multi-confirmations and DNS graph mining for malicious domain detection. *Intell. Comput.* 639–653.
- Menon, A., 2019. Thwarting C2 communication of DGA-based malware using process-level DNS traffic tracking. In: Proceedings of the 7th International Symposium on Digital Forensics and Security (ISDFS), pp. 1–5. doi:[10.1109/ISDFS.2019.8757555](https://doi.org/10.1109/ISDFS.2019.8757555).
- Nar, K., Sastry, S.S., 2018. An analytical framework to address the data exfiltration of advanced persistent threats. In: Proceedings of the IEEE Conference on Decision and Control (CDC), pp. 867–873. doi:[10.1109/CDC.2018.8619834](https://doi.org/10.1109/CDC.2018.8619834).
- E. Chien, L. OMurichu, and N. Falliere, “\$({\\$}W32. Duqu\\$){\\$}: the precursor to the next stuxnet,” 2012.
- Antonacopoulos, A., Bridson, D., Papadopoulos, C., Plutschacher, S., 2009. A realistic dataset for performance evaluation of document layout analysis. In: Proceedings of the 10th International Conference on Document Analysis and Recognition, pp. 296–300. doi:[10.1109/ICDAR.2009.271](https://doi.org/10.1109/ICDAR.2009.271).
- Koroniotis, N., Moustafa, N., Sitnikova, E., Turnbull, B., 2019. Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. *Futur. Gener. Comput. Syst.* 100, 779–796. doi:[10.1016/j.future.2019.05.041](https://doi.org/10.1016/j.future.2019.05.041).
- Skopik, F., Settanni, G., Fiedler, R., Friedberg, I., 2014b. Semi-synthetic data set generation for security software evaluation. In: Proceedings of the twelfth Annual International Conference on Privacy, Security and Trust, pp. 156–163. doi:[10.1109/PST.2014.6890935](https://doi.org/10.1109/PST.2014.6890935).
- “Machine Learning in Cybersecurity | Kaspersky.” kaspersky.com/enterprise-security/wiki-section/products/machine-learning-in-cybersecurity (accessed Jun. 30, 2022).
- “RSA NetWitness Platform Documentation - RSA Link.” community.rsa.com/t5/rsa-netwitness-platform/ct-p/netwitness-documentation (accessed Jun. 13, 2021).
- “Configuring white list mode.” support.kaspersky.com/KESWin/11.3.0/en-US/165718.htm (accessed Jun. 30, 2022).
- “Barracuda CloudGen Firewall | Barracuda Networks.” www.barracuda.com/products/cloudgenfirewall (accessed Jun. 30, 2022).
- “Advanced Malware Detection - Advanced Threat Protection | Forcepoint.” forcepoint.com/product/advanced-malware-detection (accessed Jun. 30, 2022).
- “Symantec Endpoint Protection 12.1 Business Pack - Tecdeal.” tecdeal.com/product/symantec-endpoint-protection-12-1-business-pack/ (accessed Jun. 30, 2022).
- “Preventing Multi-layered Cybersecurity Threats.” trendmicro.com/en_ie/research/21/f/proven-leadership-in-multi-layered-threat-defense.html (accessed Jun. 30, 2022).



MANAR ABU TALIB is currently teaching with the University of Sharjah, United Arab Emirates. She is also working on ISO standards for measuring the functional size of software and has been involved in developing the Arabic version of ISO 19,761 (COSMIC-FFP measurement method). She published more than 50 refereed conferences, journals, manuals, and technical reports. Her research interests include software engineering with substantial experience and knowledge in conducting research in software measurement, software quality, software testing, ISO 27,001 for information security, and open-source software. She is also the ArabWIC VP of Chapters with Arab Women in Computing Association (ArabWIC), the Google Women Tech Maker Lead, the Co-coordinator of OpenUAE Research and Development Group, and the International Collaborator with the Software Engineering Research Laboratory, Montreal, Canada.



QASSIM NASIR is a full Professor with the University of Sharjah, and the Chairman of Scientific Publishing Unit. Prior to joining the University of Sharjah, he was with Nortel Networks, Canada, as a Senior System Designer in the Network Management Group for OC-192 SONET. He was a Visiting Professor with the Helsinki University of Technology, Finland, from Summer 2002 to Summer 2009, and the GIPSA-lab, Grenoble, France, to work on a joint research project on "MAC protocol and MIMO" and "Sensor Networks and MIMO." He is also a co-coordinator with the OpenUAE Research Group, which focuses on blockchain performance and security and the use of artificial intelligence in security applications. He also conducts

research in drone and GPS jamming as well. He has published over 90 refereed conferences, journals, book chapter, and technical reports. His current research interests include telecommunication and network security, CPS, IoT, drones and GPS jamming.



ALI BOU NASSIF (Member, IEEE) received the master's degree in computer science and the Ph.D. degree in electrical and computer engineering from Western University, Canada, in 2009 and 2012, respectively. He is currently the Assistant Dean of graduate studies with the University of Sharjah, United Arab Emirates. He is also an Associate Professor with the Department of Computer Engineering and an Adjunct Research Professor with Western University. He is also a registered Professional Engineer (P.Eng.) in ON, Canada. He has published more than 65 refereed conference papers and journal articles. His research interests include the applications of statistical and artificial intelligence models in different areas, such as software engineering, electrical engineering, e-learning, security, networking, signal processing, and social media. He is a member of IEEE Computer Society.

engineering, electrical engineering, e-learning, security, networking, signal processing, and social media. He is a member of IEEE Computer Society.

TAKUA MOKHAMED received her bachelor's degree in computer science with a 3.95/4 GPA from University of Sharjah, United Arab Emirates in 2021. Currently, she is pursuing a master's degree in computer science at the University of Sharjah. She is also a Graduate Research Assistant with the University of Sharjah and the OpenUAE Research and Development Group. Her research interests include inter-blockchain communication, performance analysis of different blockchain platforms, Artificial Intelligence, Natural Language Processing (NLP) and machine learning. Takua organized a number of events and workshops for the OpenUAE research and development group.

NAFISA AHMED received her bachelor's degree in electrical and electronics engineering – software engineering department from University of Khartoum, Sudan in 2011 and received her master's degree in computer science with honor from University of Sharjah, UAE in 2019. Currently, she is a research assistant in OpenUAE Research and Development group. Her research interests include network security, Internet of Things security, and artificial intelligence. Nafisa is a mentoring and event coordinator in ArabWIC UAE chapter and an event organizer in Google Developer Group Sharjah branch.

BAYAN MAHFOOD is a member of the Research Outreach Department at the University of Sharjah. She completed her bachelor's degree in computer science with honors at the university of Sharjah in 2018 and is currently perusing a master's degree in the field of computer science at the University of Sharjah. She is a member of ArabWIC UAE chapter and Google Developer Group and has previously worked as a research assistant in the OpenUAE Research and Development Group, University of Sharjah. Her interests include Artificial intelligence and Bioinformatics.