Lin, Huaqing; Yan, Zheng; Chen, Yu; Zhang, Lifang

A Survey on Network Security-Related Data Collection Technologies

Published in:
IEEE Access

DOI:
10.1109/ACCESS.2018.2817921

Published: 01/01/2018

Document Version
Publisher's PDF, also known as Version of record

Please cite the original version:
Lin, H., Yan, Z., Chen, Y., & Zhang, L. (2018). A Survey on Network Security-Related Data Collection Technologies. *IEEE Access*, *6*(1), 18345-18365. https://doi.org/10.1109/ACCESS.2018.2817921

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 1, 2018, accepted March 14, 2018, date of publication March 21, 2018, date of current version April 23, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2817921

A Survey on Network Security-Related Data Collection Technologies

HUAQING LIN[1], ZHENG YAN[1,2], (Senior Member, IEEE), YU CHEN[3], AND LIFANG ZHANG[2]

[1]State Key Laboratory of Integrated Services Networks, School of Cyber Engineering, Xidian University, Xi'an 710071, China
[2]Department of Communications and Networking, Aalto University, 02150 Espoo, Finland
[3]Department of Informatics, University of California at Irvine, Irvine, CA 92697, USA

Corresponding author: Zheng Yan (e-mail: zyan@xidian.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB0800704, in part by the NSFC under Grant 61672410 and Grant U1536202, in part by the Project supported by the Natural Science Basic Research Plan in Shaanxi Province of China under Program 2016ZDJC-06, in part by the 111 project under Grant B08038 and Grant B16037, and in part by the Academy of Finland under Grant 308087.

ABSTRACT Security threats and economic loss caused by network attacks, intrusions, and vulnerabilities have motivated intensive studies on network security. Normally, data collected in a network system can reflect or can be used to detect security threats. We define these data as network security-related data. Studying and analyzing security-related data can help detect network attacks and intrusions, thus making it possible to further measure the security level of the whole network system. Obviously, the first step in detecting network attacks and intrusions is to collect security-related data. However, in the context of big data and 5G, there exist a number of challenges in collecting these security-related data. In this paper, we first briefly introduce network security-related data, including its definition and characteristics, and the applications of network data collection. We then provide the requirements and objectives for security-related data collection and present a taxonomy of data collection technologies. Moreover, we review existing collection nodes, collection tools, and collection mechanisms in terms of network data collection and analyze them based on the proposed requirements and objectives toward high quality security-related data collection. Finally, we discuss open research issues and conclude with suggestions for future research directions.

INDEX TERMS Network security, security-related data, data collection technologies, large-scale heterogeneous networks.

I. INTRODUCTION

With the rapid development of network and communication technologies, there has been increasing amount of attention on the security of network systems [1]. Network security is usually reflected by relevant data generated, originated or extracted from the network system. By studying the data related to network security events, the security of the network system can be quantified and measured. We refer to the data that indicates security threats and shows abnormality with regard to security, safety, privacy and trust as ***network security-related data***, in short security-related data. Obviously, the first step to detect network attacks and intrusions is to collect the security-related data.

However, there are a lot of challenges in collecting the security-related data in the current era of big data and the next generation of network systems (in short 5G). In the context of big data, the amount of data shared, originated, produced in the network is enormous. The security-related data has 5V (i.e., Volume, Variety, Value, Velocity and Veracity) characteristics, which pose tremendous difficulties in collecting these data. Further, 5G has the characteristic of being heterogeneous, supporting device-to-device, machine-to-machine and other communication technologies. In other words, 5G includes different types of networks, such as the Internet, Mobile Ad hoc Networks (MANET), mobile cellular networks and Wireless Sensor Networks (WSN), making security-related data collection difficult. Therefore, in order to evaluate the security of 5G networks, the current security-related data collection techniques in a single network need to be redesigned for large-scale heterogeneous networks. However, effective and generic security-related data collection mechanisms for heterogeneous networks are understudied. Thus, collecting the security-related data is a hot but difficult topic.

Generally, network data can be collected from both the output and input of a system and plays an important role in IT,

as it is crucial in managing and troubleshooting network system, detecting network intrusions, and billing network traffic. This paper mainly focuses on collecting data for detecting network attacks, network intrusions and network anomalies. Since network attacks and intrusions usually take place in networks, we mainly investigate data packets and data flows. Although other kinds of data, such as memory and CPU usage time, can also help detect network attacks, a previous study [2] has shown that they are not as effective as analyzing network data packets and flows for detecting network attacks and intrusions. Therefore, we do not discuss them herein. By learning and analyzing the data related to network security events, the security levels of network system can be quantified and measured, which explains the reason why collecting the network security-related data is essential. However, the definition and measurement of network security-related data are still not clear in the literature. Many network data collection methods exist in current network environments, such as sampling, similarity detecting, and traffic forecasting. Moreover, with the development of artificial intelligence, prior studies apply machine learning and deep learning to collect and process network data [2]. However, few of them are aimed at the security-related data in the era of big data and are not suitable for a large-scale heterogeneous network system [63], [64]. Furthermore, there is a lack of a thorough review on this topic yet.

In this paper, we provide a thorough review on network data collection technologies and compare existing works according to a number of functional and security objectives for the purpose of high quality security-related data collection. We then discuss open issues, challenges and future research trends in this field. Specifically, our contributions can be summarized as follows:

1) We propose a number of requirements and objectives in terms of collecting the security-related data in a large scale heterogeneous network system, which can be used to evaluate existing related works;

2) We comprehensively review the methods, mechanisms and technologies for collecting network data by applying the proposed requirements and objectives to evaluate their performance towards high quality network security-related data collection;

3) We summarize open research issues and propose future research trends in collecting the network security-related data based on in-depth literature study and analysis.

The rest of this paper is organized as follows. Section 2 gives a brief introduction on network security-related data collection. Section 3 introduces the requirements and objectives of network security-related data collection. In Section 4, we thoroughly review existing network data collection technologies and discuss if they are qualified for collecting the security-related data. In Section 5, open issues, challenges, and future research trends in collecting the network security-related data are presented. Finally, the paper concludes in the last section.

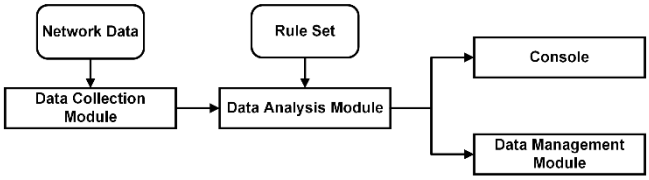


FIGURE 1. A basic model of IDS.

II. NETWORK SECURITY-RELATED DATA COLLECTION

In this section, we define network security-related data and summarize the main purposes and applications of network data collection.

A. DEFINITION OF NETWORK SECURITY-RELATED DATA

Usually, data anomalies can reflect network attacks and intrusions. Thus, we can detect network attacks by searching for abnormal network data. We refer this kind of data that can reflect the security status of a network system as ***network security-related data***, and they may be feature, signature or fingerprint of a specific attack behavior [3]. For example, Time To Live (TTL) is a type of network security-related data. TTL specifies the maximum number of segments allowed to pass before an IP packet is discarded by a router. We consider TTL as security-related data about network packets because it can be used to detect Denial of Service (DoS) attacks. Abnormal TTL signals high probability that the network system is being intruded. For example, an unusually excessive number of packets with the same TTL entering a network simultaneously implies that a DoS attack is likely occurring. Thus, many security applications (e.g., Intrusion Detection Systems - IDS) monitor TTL in order to detect DoS attacks.

B. PURPOSES AND APPLICATION SCENARIOS

Collecting network data is becoming increasingly important, particularly with the flourishing of big data and Internet of Things (IoT). The purposes of collecting network data mainly include 1) intrusion detection, 2) network management, 3) traffic accounting, 4) network forensics, and 5) malware detection. We introduce the purposes and application scenarios of network data collection in more detail below.

1) INTRUSION DETECTION

Intrusion detection refers to the behavior of monitoring and detecting malicious activities or policy violations in a network or system. The most used scenarios of network security-related data collection are IDS and other network security systems or security devices that detect network attacks and intrusions. The data collection module of an IDS is responsible for monitoring the host state, the network data and user behaviors. Network data here include the parameters of network activities, the number of network connections, the number of packets, the content of packages, etc.

As shown in Fig. 1, the data collection module is an important component of the IDS model. Data collection is easy

to become the performance bottleneck of IDS, because the efficiency of data collection module in IDS directly affects the performance of intrusion detection. As a result, data collection is crucial with regard to the performance of IDS [4], [5]. Prior studies used various ways to collect security-related data for intrusion detection. For example, Shyu *et al.* [6] proposed an unsupervised anomaly detection schema based on Principle Component Analysis (PCA). Based on the work of Ramah *et al.* [4] and Shyu *et al.* [6] implemented a system that can detect anomaly traffic automatically in real time. Fessi *et al.* [7] proposed an architectural distributed IDS where collectors are scattered in a network system in order to monitor and collect network data. This scheme can detect attack scenarios by performing internal correlation. Singh and Joshi [8] used a honeypot to masquerade as a normal system to collect network intrusion and attack data. The honeypot captures the security-related data to record attacks and intrusions. Some efficient sampling mechanisms are proposed to improve the performance of data collection in IDS [9], [10]. Yan *et al.* [70] collected security-related data to conduct trust evaluation on network entities and perform trust management in order to do spam detection [68], intrusion detection [69], and unwanted traffic control [70].

2) NETWORK MANAGEMENT

Network management includes network diagnoses, fault detection, network configuration and design. A Network Management System (NMS) troubleshoots network fault, performs network configuration, and monitors quality of service. An efficient NMS focuses on collecting real-time network data in order to monitor relevant resources and network performance, thus guaranteeing the efficiency and robustness of the network system. Network data collection has long been the central component of NMS or Network Protocol Analyzer. Improved technologies can make network data collection more suitable for debugging, optimization, measurement, profiling and detection of network system anomalies. For instance, Ji *et al.* [11] introduced a network data collection model based on feedback according to the data correlation for effective network management. Ahn and Chae [12] designed and implemented a network traffic monitoring system for the network management of a PC-room. By monitoring network traffic, the proposed system can effectively realize network management, which includes network fault detection and network configuration. Falaki *et al.* [13] collected network traffic in smartphone and analyzed its characteristics in order to further improve the efficiency of the network system and other applications. By studying the relationship between traffic and radio power, they built a model to reduce the power consumption of the radio.

3) TRAFFIC ACCOUNTING

Traffic accounting means that Internet service providers charge users who subscribe to their Internet access services according to some policy. Normally, the Internet access service is charged. Therefore, the Internet service providers (ISPs) and the intranet of enterprises and companies need their own traffic accounting systems to charge their subscribers. A typical traffic accounting system is composed of traffic collection, accounting information statistics, accounting rule configuration and user management. However, current commercial traffic accounting systems cannot meet diverse customer requirements with respect to system functionality. The prices set by these systems are relatively high. Thus, it is practical for the customers to implement their own traffic accounting system to satisfy their own functional requirements. Li *et al.* [14] introduced several network traffic collection techniques that can be used in accounting systems, such as router-oriented, proxy server-based, firewall-oriented and Ethernet-LAN-oriented.

4) NETWORK FORENSICS

Network forensics is often applied to finding out network anomalies, detect intrusions and attacks, and track down cyber-crimes [15]. Network forensics also necessitates traffic data collection, since it can provide raw data to analyze and reveal valuable information [16]. For example, Parry *et al.* [17] points out that any forensic investigator needs to have the ability to collect the required data and find important and critical information from collected data in order to understand an anomaly. Therefore, efficient data collection becomes the basis of network forensics.

5) MALWARE DETECTION

Malware refers to the software with malicious behaviors running in a host or a network system [67]. Their malicious behaviors include damaging a system, collecting personal information, attacking or intruding a network, etc. Many kinds of malware control malicious behaviors and operations through networks. As a result, during the process of conducting malicious behaviors, malware generates abnormal network data packets. Therefore, analyzing network traffic can help detecting malware [18], [19]. In particular, with the development of the mobile Internet, increasing amount of mobile malware has appeared. The mobile malware can be detected by analyzing data packets of mobile communications, such as Domain Name System (DNS) queries and Address Resolution Protocol (ARP) requests. For example, Han *et al.* [19] proposed to identify malicious mobile applications by analyzing malware traffic in the mobile Internet and achieved a high detection rate and scalability in their experiments.

6) OTHER PURPOSES

The above five kinds of network data collection purposes are the most common. However, there are several other purposes as we briefly describe below. For example, testing and evaluating the effectiveness of data mining and machine learning algorithms requires the collection of normal and abnormal network datasets (e.g., KDD99) [20]. Detecting malicious domains is another scenario that needs to collect

network data. For example, monitoring DNS traffic can help detect and recognize malicious domains [19]. Collecting network data is also essential for hackers to prepare network attacks and intrusions. However, our goal is to make network and system secure. Thus, data collection is a double-edged sword.

III. REQUIREMENTS AND OBJECTIVES OF SECURITY-RELATED DATA COLLECTION

In this section, we present the requirements and objectives in terms of network security-related data collection. Based on the current literature and research, the requirements mainly include *functional* requirements and *security* requirements and the objectives consist of *functional* objectives and *security* objectives. According to the requirements and objectives, existing works on security-related data collection can be evaluated with respect to function and security.

A. REQUIREMENTS

1) FUNCTIONAL REQUIREMENTS (FR)

The functional requirements (FR) are those that must be implemented in order to collect network security-related data. We list the requirements based on the current literature as below.

- FR1: Must be able to collect required security-related data in different situations and contexts [21].
- FR2: Must be able to store collected data in a storage medium [22].
- FR3: Must be able to know where and when to collect security-related data [22].
- FR4: Must be able to load information about which data should be collected [22].
- FR5: Must be able to export data to other systems or create external database [22].
- FR6: Must have the ability to manage and control data [23].
- FR7: Must be efficient and stable when collecting data [24].
- FR8: Must be flexible and scalable in data collection [25].
- FR9: Must not cost too much computation resources, storage resources or other resources in collecting data in some scenarios [21].
- FR10: Must be automatic and adaptive, with a certain degree of intelligence and learning ability in order to adapt to the changes of a network environment [11].
- FR11: Must not destroy the original network system [24].
- FR12: Must be universal and generic and be able to support a variety of application scenarios [26].
- FR13: Must not produce new data that might affect the accuracy of collected data [16].

2) SECURITY REQUIREMENTS (SR)

The security requirements (SR) are those that deal with quality and security issues when collecting network security-related data.

- SR1: Must be able to prevent data loss and ensure data truth in collecting data (data integrity, veracity and availability) [22].
- SR2: Must protect user privacy in collecting data [27].
- SR3: Must ensure the security of collected data and be able to prevent data leakage [22].
- SR4: Must be able to verify the integrity and authenticity of the collected data [22].
- SR5: Must have access control capability that can authenticate users who want to access data and authorize the access for eligible users [27].

TABLE 1. Relationship between objectives and requirements.

Type	Objectives	Requirements
Functional	Applicability	FR1, FR4
	Adaptability	FR10
	Scalability	FR8
	Stability	FR7
	Generality	FR12
	Flexibility	FR8
	Efficiency	FR7
	Non-destructivity	FR11, FR13
	Cost	FR9
Security	Confidentiality	SR3, SR5
	Integrity	SR1, SR4
	Non-repudiation	SR4
	Authentication	SR4, SR5
	Privacy Protection	SR2
	Self-protection	SR1, SR3, SR5

B. OBJECTIVES

Based on the above functional requirements and security requirements, we propose several objectives need to be achieved in the process of security-related data collection. The relationship between the objectives and the requirements is shown in Table 1. It must be noted that objectives of network security-related data collection are different from the objectives of general network data collection. In Section IV, we use the proposed objectives of network security-related data to evaluate the existing work about network data collection.

1) FUNCTIONAL OBJECTIVES

a: APPLICABILITY (APP)

Based on FR1 and FR4, we propose an objective named *Applicability*. Applicability in this paper refers to that a proposed network data collection technique or mechanism can be deployed into a real network environment to collect network security-related data.

b: ADAPTABILITY (ADA)

Based on FR10, we propose an objective named *Adaptability*. Adaptability refers to that a collection mechanism can adjust

to different network contexts and situations. For example, the collected content can be chosen according to network context variation or the collecting frequency can be adjusted based on network data variation.

c: SCALABILITY (SCA)

Based on FR8, we propose an objective named *Scalability*. Scalability refers to the quality of being scalable for a network security-related data collection technology. A scalable data collection technology should be based on a scalable architecture to support data collection of different types and various volumes of data. In this way, a scalable collection technology can be expanded in order to adapt to evolving network systems.

d: STABILITY (STA)

Based on FR7, we propose an objective named *Stability*. Stability refers to the fact that a network security-related data collection system, mechanism or algorithm should not be abnormal (e.g., buffer overflow and thread deadlock) in the course of their operation. In addition, both the hardware and software used for data collection should be stable and system downtime should not occur frequently.

e: GENERALITY (GEA)

Based on FR12, we propose an objective named *Generality*. Generality refers to that network security-related data collection technologies can be applied to multiple scenarios and can collect different types of data. A network security-related data collection technology with good generality can simplify a collection model, reduce the cost of use and enhance the capacity of data collection.

f: FLEXIBILITY (FLB)

Another objective named *Flexibility* can be proposed based on FR8. Flexibility refers to that a network security-relate data collection mechanism is switchable between different network contexts and reserves a reasonable number of functional extension interfaces. The flexibility is demanded by the upcoming large-scale and heterogeneous network. A flexible data collection mechanism should be able to solve heterogeneous problems between various types of networks in the context of large-scale networks.

g: EFFICIENCY (EFE)

Based on FR7, we propose an objective named *Efficiency*, which refers to that a data collection technology can collect required data efficiently without affecting normal network system operations and performance. Especially for a large-scale high-speed network, an efficient network data collection technology is much needed.

h: NON-DESTRUCTIVITY (N-DES)

Based on FR11 and FR13, we propose an objective named *Non-destructivity*. Non-destructivity means that a collection technology cannot destroy the functionality of actual network devices, such as switches and routers, from the system point of view. In addition, it cannot alter normal communication data, for example, by introducing irrelevant data, from the communication data point of view. Because the normal operation of a network system and the credibility of collected data must be ensured, non-destructive data collection is very important. According to the performance of a data collection method, we can evaluate the non-destructivity based on three levels: Low means that the deployed data collection method greatly destroys the functionality of the application system or introduces a lot of useless data and thus affect the authenticity of the collected data; Medium means that the deployed data collection method slightly destroys the functionality of the application system or introduces some useless data; High means that the deployed data collection method does not destroy the functionality of the application system or does not introduce useless data.

i: COST (C)

Based on FR9, we propose an objective named *Cost*. As a practical technology, data collection must take its cost into consideration. The cost must fit its application scenarios. For example, in common scenarios, data collection should be lightweight and thus cost little. However, some important scenarios (such as those related to national defense or military) necessitate more complex data collection techniques with high performance, allowing higher costs. According to the cost to deploy a data collection method, we can divide its cost into three levels: High means that the cost of deploying the data collection method is high, which is generally adopted for the applications that require high performance; Medium means that deploying a data collection method requires a bit of cost, but it is reasonable and acceptable in most cases; Low means that the deployment of the data collection method does not need any cost for common data collection scenarios.

2) SECURITY OBJECTIVES

a: CONFIDENTIALITY (CFD)

Based on SR3 and SR5, we propose an objective named *Confidentiality*. Confidentiality means that data or information cannot be disclosed to an unauthorized party and is usually guaranteed by cryptographic systems (e.g., AES or ECC). Normally, encrypted data is not understandable anymore unless they are decrypted. Thus, encryption can protect collected data from leaking even if they are lost. The premise, of course, is that the encryption algorithm is secure enough. Therefore, collected data cannot be stored in the form of plaintext and should be encrypted in order to ensure confidentiality if they are valuable or contain sensitive information of some party.

b: INTEGRITY (IT)

Based on SR1 and SR4, we propose an objective named *Integrity*. Integrity means that data or information cannot be lost, modified by attackers or replaced by other

erroneous data. The integrity of collected data must be guaranteed and the data cannot be lost. Because once some part of data is lost, the reliability of the collected data cannot be guaranteed anymore. The cryptographic techniques such as MD5 or SHA can ensure that once data is lost, it can be found immediately. Digital digest is one of the most commonly used methods to ensure data integrity.

c: NON-REPUDIATION (N-RP)

Based on SR4, we propose an objective named *Non-repudiation*. Non-repudiation refers to the non-repudiation of one's collection behaviors and the non-repudiation of the time and location of these behaviors. Non-repudiation of collection behaviors here mainly refers to that collecting nodes cannot deny the fact that they have collected data at some time and in some place, and that the sourcing nodes that provide data cannot deny the fact that the collected data were from them. Only when the non-repudiation of collected data and collecting nodes is ensured, can we track data collecting process.

d: AUTHENTICATION (AUT)

Based on SR4 and SR5, we propose an objective named *Authentication*. Authentication refers to that both sides of data collection in collecting security-related data need to carry out identity authentication. Authentication can help validating and guaranteeing the authenticity of collecting node identities and the credibility of collected data, thus preventing collecting data from a malicious node.

e: PRIVACY PROTECTION (PP)

Based on SR2, we propose an objective named *Privacy Protection*. Privacy protection refers to that collecting private information should be avoided or the privacy information contained in collected data need to be anonymized and their confidentiality should be assured, so as to prevent privacy from being leaked. Only when a data collection process has sufficient privacy protection capability, can it be accepted. Moreover, privacy legislation requires network data collection techniques to protect privacy [62]. Thus, privacy protection is a must, not an option.

f: SELF-PROTECTION (SPT)

Based on SR1, SR3 and SR5, we propose an objective named *Self-protection*. Self-protection mainly refers to protecting collected data and preventing a collection system from being destroyed. The collected security-related data should have a self-protection mechanism that can ensure data integrity and prevent data from leaking by either encrypting or authenticating. The collection system should have self-protection capability to prevent the operating system from being destroyed by external attackers.

IV. NETWORK DATA COLLECTION TECHNOLOGIES

In this section, we survey the technologies that are applied in collecting network security-related data. These technologies relate to *collection nodes*, *collection tools*, and *collection mechanisms*. We find that a complete security-related data collection scheme first need to determine collection nodes, i.e., the location of data collection, and then decide collection tools. Finally, it is important to design a collection mechanism, i.e., a collection control strategy or adjustment algorithm. Herein, we present a revised taxonomy that classifies network data collection technologies based on collection nodes, collection tools and collection mechanisms. We also discuss the pros and cons of existing data collection mechanisms in dealing with the requirements and objectives described above.

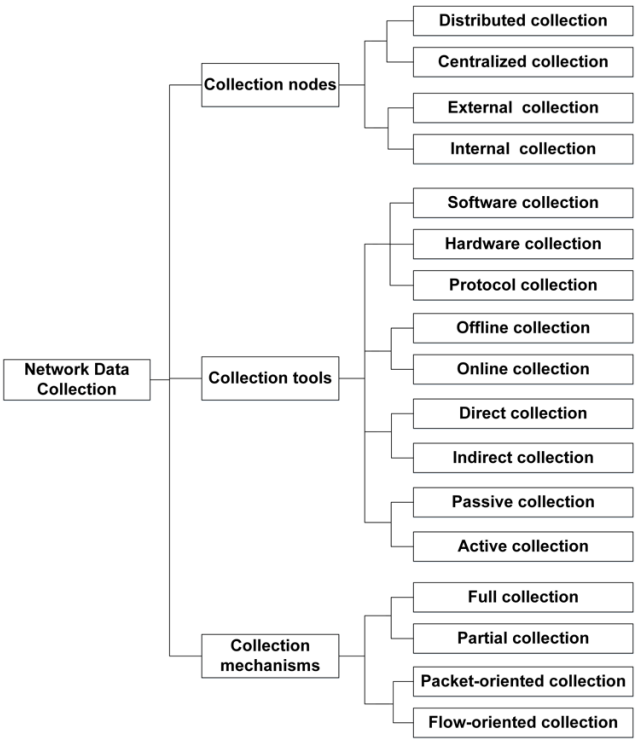


FIGURE 2. A revised taxonomy of network data collection.

A. A REVISED TAXONOMY

We derive here a taxonomy about network data collection technologies from existing related works in order to provide a useful overview of them. There are a lot of taxonomies for data collection techniques proposed from various aspects in previous works. However, currently there does not exist a uniform and accepted taxonomy for data collection technologies. In this paper, we focus on three aspects of the data collection technologies: collection nodes, collection tools and collection mechanisms, and classify the existing data collection technologies accordingly. Fig. 2 presents our proposed taxonomy of network data collection technologies. This revised taxonomy is comprehensive, complete and clear, covering almost all data collection technologies. It helps us understanding the network data collection technologies and further guides us in selecting or designing a suitable data collection method in a concrete application scenario. It is worth mentioning

that each data collection technology may belong to multiple categories.

First, the data collection technologies can be classified according to collection nodes. Based on the number of data collection nodes, data collection technologies can be divided into centralized data collection and distributed data collection. The centralized data collection method can facilitate the management and coordination of collection nodes, but the collected data type is unitary. The distributed data collection can collect multiple types of data to comprehensively understand the status of an entire network system. But it is not easy to coordinate various distributed collection nodes. Based on the location of data collection, data collection technologies can be divided into internal data collection and external data collection. The internal data collection can collect the communication data inside a system, but it may affect the normal operation of the system. External data collection can only collect external communication data, but it does not burden the system.

Second, we can classify the data collection technologies based on collection tools. Based on the used collection tools, data collection technologies can be divided into software-based data collection, protocol-based data collection and hardware-based data collection. The software-based data collection has high flexibility and low cost, but low performance. The hardware-based data collection is suitable for the application scenarios that require high performance, but its cost is high. The protocol-based data collection can collect various types of data, thus providing an overall understanding of an entire network system, but it is not very flexible in terms of data collection. Based on whether data is collected online, data collection technologies can be divided into online data collection and off-line data collection. The real-time performance of the online data collection is high, but it could burden the network system. While the off-line data collection performs oppositely. Based on whether data is collected directly, data collection technologies can be divided into direct data collection and indirect data collection. The direct data collection can collect many types of data with high accuracy, but it could burden the system, while the indirect data collection performs differently. Based on whether it is initiative to collect data, data collection technologies can be divided into active data collection and passive data collection. The active data collection can collect more data than the passive data collection. However, some active data collection technologies may not provide accurate data.

Third, we can also classify the data collection technologies based on collection mechanisms. Based on the amount of data collected, the data collection technologies can be divided into partial data collection and full data collection. The partial data collection can reduce the amount of data collected, thus avoiding its burden on the system. The data collected by the full data collection is more accurate. But it brings extra burden on the system. Based on the format of data collected, the data collection technologies can be divided into packet-oriented data collection and flow-oriented data collection. The packet-oriented data collection is used to collect data in the form of data packets. The flow-oriented data collection, however, collects flow information and is dependent on routers with flow collection functions. Notably, the above taxonomy is cross, and some data collection technologies may belong to multiple categories. In what follows, we review the current state of art in terms of collection nodes, collection tools and collection mechanisms of network security-related data collection. In our review, the terms ''method'' and ''technology'' may be used by replacing with each other, depending on a concrete description context, but they have the same or similar meanings.

B. COLLECTION NODES

In a network system, common data collecting nodes include routers, switches, gateways, IDSs/IPSs, firewalls, honeypots, sensors, proxy servers/collecting servers, agents, mobile terminals, and distributed collection nodes. The routers, switches and gateways are components of the Internet, whose main function is delivering packets. The IDSs/IPSs, firewalls and honeypots are specially used network equipment for security purposes. The sensors, proxy servers/collecting servers and agents are specially designed equipment for data collection. Mobile terminals including smartphones, tablets, and wearable smart devices are usually network terminal devices. Due to their mobility and flexibility, they are usually used to collect network data.

C. COLLECTION TOOLS

There are various ways to classify network data collection tools. In this paper, we focus on the most common ones, which categorizing collection tools by software-based collection tool, hardware-based collection tool and network protocol-based collection tool. Normally, collecting data with hardware equipment has high performance, thus it is suitable for large-scale network systems. However, it costs a lot and has the disadvantage of being inflexible and not universal. Network protocol is another way used by network administrators to collect data. However, this method is not applicable for host terminals and is too complicated for mobile devices. Therefore, a universal security-related data collector that can be applied at different network nodes (e.g., routers, switches, network servers, PC hosts or mobile devices) is needed. Moreover, the collector should be applicable to both a single independent system and heterogeneous network systems. Next, we introduce several different data collection tools in detail.

1) SOFTWARE-BASED DATA COLLECTION

Packet capture based on software consists of several subsystems, as shown in Fig. 3. The network card, device driver, the capturing stack of operating system, packet capture library and packet capture application are involved in packet collection and processing. A problem occurring in any one of the subsystems leads to packet loss, thus yielding bad capturing results. In order to achieve high efficiency and realize

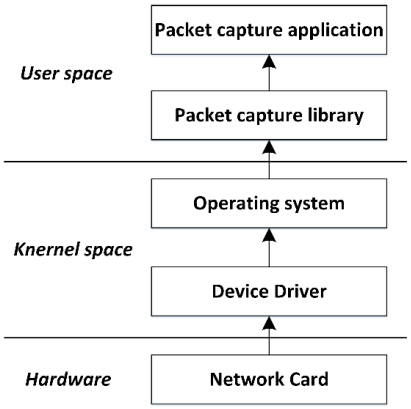


FIGURE 3. The subsystems involved in packet capture based on software.

a specific function, it is necessary to expand and improve the packet capture library, device driver, capturing stack and network card.

a: LIBPCAP

Libpcap is a commonly used packet capture library. It is found to be the basis of sniff software such as tcpdump and snort [5]. It is worth noting that most of the data collection systems use pcap format for storage [28]. This is because it is the standard format for documenting network data packets.

b: IMPROVED LIBPCAP

Unfortunately, libpcap incurs many interrupts and repeated replication between memories when collecting data packets. To overcome such problems, many efforts have aimed to improve libpcap [29]. For example, Woods proposed to use a shared memory to exchange network data packets between kernel space and user space in order to avoid a large number of replication and interrupt operations, thus improving packet capture efficiency [66]. Papsadogiannakis *et al.* [30] introduced a technique called subzero packet copy that can avoid copying uninteresting packets across different memory areas and another technique named prioritized packet loss that can be adapted to overload conditions by dropping the packets with lower priority. Moreover, they proposed a stream-oriented network monitoring library named Stream capture library (Scap) based on the proposed two technologies. These two techniques realize efficient stream collection.

c: IMPROVED DRIVER AND NETWORK STACK

The drivers, capture stacks of an operating system, and monitoring applications all need to be improved to take full advantage of available hardware in order to achieve the best packet capturing performance [29]. Deri and Fusco [31] tried to improve the capturing performance by modifying drivers from two aspects. First, they spawned a thread and dedicated it to packet consumptions in the driver. Second, they reused Direct Memory Access (DMA) memory page for Network Interface Cards (NICs). In [32], nCap was proposed to capture packets in wire-speed. Instead of using standard packet processing software, nCap modified drivers to create two circular buffers for setting incoming and outgoing packets and improved a capturing library to allow capture applications to read packets directly from the NICs.

d: SIMULATION SOFTWARE

Collecting network datasets requires simulation of a real network context [20]. Normally, in order to reduce the cost, we usually use Software-Defined Network (SDN) [33] or some network system simulation software (e.g., NS3) [34] to simulate the real network functions. Collecting data for analyzing and processing based on a simulated network environment is applied in many scientific researches in recent years.

2) NETWORK PROTOCOL-BASED DATA COLLECTION

Network protocol-based data collection technologies can provide a comprehensive understanding of the whole network system and are usually applied in the application scenarios of network management and network problem diagnosis.

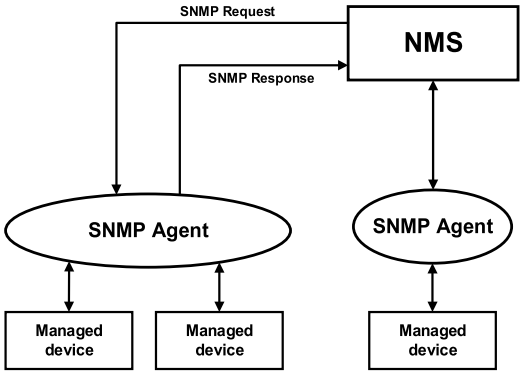


FIGURE 4. Principle components of SNMP Communications.

a: SNMP

Simple Network Management Protocol (SNMP) is a network protocol widely used to monitor, control, configure and manage elements in network systems, such as network devices [14]. A network deployed with SNMP consists of three key components: network device, SNMP agent and NMS, as shown in Fig. 4. With the help of SNMP, network managers can locate and troubleshoot problems such as network failure. SNMP, which is supported by almost all routers, can be used for communications between network managers residing in NMS and SNMP agents residing in network devices such as routers and switches [14]. Moreover, in order to derive analysis information of a real-time request message, the SNMP can conduct relevant Management Information Base (MIB) information polling [12].

Ramah *et al.* [4] utilized SNMP to periodically collect network traffic from the MIB of a central firewall in a campus network, which is analyzed to detect anomaly. Garcia-Dorado *et al.* [35] also applied SNMP to periodically poll an interface table in order to collect traffic traversing a router.

b: TELNET

Telecommunication network (Telnet) protocol is another network protocol that can be used to capture network data packets [14]. A Telnet client can be configured to periodically setup telnet connections with a router. And the router will return text-based results to the client, which can be analyzed to extract traffic data. The speed to collect traffic with the Telnet protocol is comparatively faster than that of SNMP protocol. Moreover, Telnet-based data collection technology only results in minimal data redundancy. However, Telnet needs root authority to collect network data, which could cause potential network security problems [14].

c: IPFIX

IP Flow Information Export (IPFIX) is a common, universal standard that defines how IP flow information is to be formatted and transferred from routers, probes and other devices to a collector for network measurement or traffic billing [27]. IPFIX offers a significant degree of flexibility in letting users to define data types they need to convey. Some probes export flow-data using IPFIX protocols [27]. IPFIX protocols are often used to format preprocessed reports in some traffic monitoring applications to address the difficulty of identifying and detecting distributed network anomalies and attacks. Moreover, IPFIX protocol can collect data at multiple scattered nodes of a network system. In this way, it can provide a comprehensive and objective understanding of the network system.

d: NETFLOW

NetFlow, which is a data exchange mode introduced by Cisco, is mainly used for planning network traffic and managing traffic growth [20]. In addition, Cisco further provides Cisco NetFlow Collector (NFC) in order to collect NetFlow data. Many other manufacturers also offer similar collection software. From the point of view of capturing network traffic, NetFlow can provide a higher level of abstraction for assembling related packets into groups, which are referred as flows [20]. Alias *et al.* [36] combined the NetFlow protocol and device polling to design an enhanced passive packet capture scheme that can capture and analyze packets in a Gigabit network.

3) HARDWARE-BASED DATA COLLECTION

Hardware based data collection technologies are commonly used in application scenarios that require high performance such as IDS. But the cost of this approach is high. Schneider *et al.* [37] found out that the AMD platform yielded better capturing results than the Intel platform by comparing capturing hardware based on Intel Xeon CPUs and AMD Opteron CPUs with similar components. The most reasonable explanation is that AMD can manage memory and handle bus contention better than other products. We next introduce other hardware devices that are used for data collection.

a: SENSOR

Sensors are commonly used data collection tools, which have high collection efficiency and have special characteristics and properties. For example, sensors are very flexible with regard to data collection and other hardware is not. Gad *et al.* [25] proposed an approach to flexibly capture distributed remote packet based on sensors with additional self-adaptability and cooperation capabilities.

b: HARDWARE PROBE

Hardware probe is a network tool used to monitor network packets and also has the function of filtering and analyzing [38]. Hardware probe can provide the complete information of real-time traffic from the physical layer to the application layer, without affecting the performance of a network system. Gao *et al.* [38] designed a hardware probe to collect real-time traffic on a network link. Bonelli *et al.* [27] designed and implemented a smart probe that can support traffic pre-processing according to the needs of specific applications in a scalable and performance-effective manner. This approach has two advantages. First, it allows to discard a huge amount of irrelevant information, thus relieving the burden to an application. Second, the traffic pre-processing can be used to protect privacy. Moreover, the smart probe proposed in [27] can interact with external networks using strict role-based policies, making it easy to integrate with a standard access control infrastructure.

c: DAG CARDS

Data Acquisition and Generation (DAG) cards are data capture cards that were designed to capture network packets [17]. Normally, they are especially effective in capturing packets in large-scale high-speed networks. And their goal is to capture 100 percent packets in any networks regardless of package size, interface type, or network load. DAG based network traffic capture technologies have been proven to have the most accurate visibility of network systems. Moreover, DAG cards can provide a range of additional hardware-based functions such as load balancing, packet filtering and classification, traffic replication and time stamping, further providing higher performance than a software-based solution. As a result, DAG cards have become an industry standard for network security monitoring, leading to many existing researches on DAG cards based data collection technologies [17]. For example, in order to inject traffic bursts, Zabala *et al.* [39] installed an Endace 4.3GE DAG card in an injector machine.

However, DAG cards provide only limited on-board filtering functionalities [27]. In addition, even though the packet capture rate of DAG cards has been proved to be 100 percent accurate, it is still a challenge to fully utilize it due to its higher cost and lower flexibility of hardware compared to software [36].

d: PORT MIRRORING

Port mirroring can be deployed at the switches, routers or gateways that have a port forwarding function and is used to

send a copy of network packets seen in one switch (or router) port to a network monitoring connection in another switch (or router) port [26]. It is a common method to collect network traffic. When a user device connects to the Internet through these devices on one port, traffic data are mirrored into the collecting server via a mirroring port [19]. Either inbound or outbound network traffic exported from a mirroring port can be aggregated and collected using tshark or tcpdump [20].

The advantages of port mirroring are that they are simple to install and can be activated as needed without affecting the existing network system. However, port mirroring also have several disadvantages [17]. First, a mirroring port may drop malformed packets or stop mirroring packets altogether if the computation buffers of switches or routers is full. In this case, the collected data is not complete and thus inaccurate. Second, port mirroring has to be configured to recognize Virtual Local Area Network (VLAN) traffic. Moreover, switches or routers that have port mirroring may inject their own packets into network systems and may modify the priority of existing packets. As a result, port mirroring may not be passive.

e: INLINE TAPS

Inline tap is a hardware port that provides a full view of the network without any impact on the network system and network data. Compared to port mirroring, inline taps has the advantage of offering greater certainty that all network traffic is collected, while port mirroring may drop some packets as described above [17]. On the other hand, installing an inline tap requires breaking the communications link resulting in many difficulties which is not as easy to install and configure as port mirroring.

f: NETWORK INTERFACE CARD

The use of the underlying NIC is inevitable for any kinds of traffic capture technologies. Network cards and corresponding software applications can be customized in order to achieve special purposes [2], [26], [29], [30]. Unfortunately, the capturing tools relying on standard NICs have several limitations. First, they are potentially noisy. Second, they cannot preserve packets orders. Third, they lack the capacity to time precisely. In order to overcome these limitations, Parry *et al.* [17] combined hardware and software solutions based on a modified dedicated NIC and implemented a collection system that can guarantee to collect all data packets.

g: MOBILE TERMINAL

Mobile devices are playing an increasing important role in our daily life. For example, increasing number of researchers use mobile terminals (e.g., smartphones and tablets) to collect data [68]–[70]. Ariyapala *et al.* [21] utilized smartphone to design a system to detect network attacks and anomalies where the smartphone was used to capture the network traffic, network logs and the system logs. Boualouache *et al.* [23] proposed to use smartphones as data collectors for their IoT data collection system. And it can forward the collected data from the data collector to the data gateways based on Bluetooth Low Energy (BLE) technology. Jiewu *et al.* [41] also applied mobile terminals to collect real-time traffic and network logs of cellular subscribers.

h: IDS/IPS

IDS/IPS based on hardware is an independent network security device used to detect network intrusions and anomalies [5], [42]. It detects network attacks and anomalies by collecting network data in a network system. There are two kinds of intrusion detection technologies: anomaly detection and misuse detection. In recent years, with the development of information technology and the improvement of security requirements, there has been increasing amount of work on IDS/IPS [5].

i: FIREWALL

A firewall based on hardware is a network security device deployed in the boundary points between intranet and extranet in series that monitors the incoming and outgoing network traffic based on rules set in advance. Firewalls are appropriate collecting nodes when collecting traffic through a network system is needed [14]. Currently, there are three categories of firewalls: Packet Filter firewalls, Application layer firewalls and Proxy-based firewalls.

Packet Filter firewalls inspect the fields in a packet header (e.g., the destination IP address or destination port of a packet) to determine whether the packet is allowed to pass or discard. Application layer firewalls can identify certain applications and protocols such as FTP and HTTP by Deep Packet Inspection (DPI) or Deep Flow Inspection (DFI). Proxy-based firewalls utilizes proxy servers to separate an external network from an internal network and provide authentication, logging and account management functions.

It is preferable to integrate a collecting function into the firewalls in order to collect network data from both inside and outside networks and thus can detect attacks and anomalies originating from both of them.

j: PROXY SERVER

Some Local Area Networks (LANs) utilize proxy servers to carry out network relaying, thus enabling data collection through them [14]. Currently, most proxy servers can record all transmitted data in detail and store them to disk as text-based traffic logs. Some of them can even store the log files into a database. Thus, the proxy server can be a practical data collection tool.

k: AGENT

Fessi *et al.* [7] proposed a system which uses a hierarchical structure to collect the information produced by multiple agents running in different hosts, so as to detect anomalies and attacks. This method of using multiple agents to collect network data is suitable for detecting distributed network attacks.

l: HONEYPOT

A honeypot is an information security system which can be disguised as a real information system, thereby diverting network attackers away from critical information systems resources [8], [43]. It is also a tool for studying the attack behavior by recording abnormal information. The honeypot system can be used to collect known or unknown attack data in order to study abnormal behaviors to avoid critical systems being attacked [8]. Moreover, honeypots can work in an encrypted network environment, which is not available to other network security devices. However, deploying a honeypot system is very expensive and requires careful configuration to achieve the purpose of defense attacks and record attack information.

D. COLLECTION MECHANISMS

In this part, we review existing network data collection mechanisms and comment their advantages and disadvantages in a general way. Herein, we focus on partial data collection and full data collection since it is a good taxonomy covering all data collection technologies. In the next sub-section, we will further analyze whether they are qualified to be applied to collecting network security-related data according to the proposed requirements and objectives.

1) PARTIAL DATA COLLECTION

a: TRAFFIC PREDICTION BASED DATA COLLECTION

With the development of artificial intelligence and machine learning, demands for intelligence are becoming increasingly high [2]. This is because machine learning based data collection methods such as traffic prediction can help an Access Point (AP) to well perform access control, load balancing, and QoS assurance.

In traditional wireless networks, one monitor is usually responsible for collecting traffic on one specific channel. This method incurs high cost because typically a cognitive radio network has a large number of channels. Chen *et al.* [15] utilized only a small number of monitors to collect data in cognitive radio networks by predicting packet arrival time with incremental Support Vector Regression (SVR) and then intelligently switching monitors between multiple channels. They also proposed to schedule multiple monitors in order to scan channels and capture packets effectively. Simulation results show that their packet capture rate was higher than 70%, which is much better than a random scheme.

Jiewu *et al.* [41] proposed to collect network data based on a SVR and MapReduce framework. The MapReduce is able to improve the computing power and scalability of the system architecture. On the other hand, SVR can flexibly predict traffic for its remarkable generalization performance. Wen *et al.* [44] applied spatial-temporal compressed sensing to predict network traffic . Yu *et al.* [45] proposed to predict 3G traffic with multiracial exploration. Wang and Shan [46] proposed to predict traffic with wavelet since wavelet is a natural way to describe the multi-scale characteristics of self-similarity of network traffic. Krithikaivasan *et al.* [47] utilized Autoregressive Integrated Moving Average (ARIMA) model to predict network traffic. Artificial Neural Network (ANN) was often proposed to predict traffic for the non-linear nature of network traffic. Theoretically, ANN can capture any kind of relationship between the output and the input [48]. However, it might suffer from overfitting [49]. Recently, Support Vector Machine (SVM) has been successfully applied in many areas, such as prediction in time series [50].

b: SAMPLING BASED DATA COLLECTION

Currently, all IDSs are trying to collect the whole traffic in a network. However, it is inevitable for IDS to miss some packets due to the limitations of computer resources such as the lack of storage, computation and memory. This is especially true in a large-scale high-speed network with heavy traffic. Thus, it is impossible for an IDS to collect real-time network data completely. In addition, the proportion of network traffic with attack signature is relatively small. Moreover, capturing the whole network traffic degrades the utilization rate of network bandwidth and downgrade the performance of a network application system. Therefore, IDSs need a more reasonable network traffic collection technique. Sampling is one of such techniques to realize collecting network traffic reasonably for IDS, especially in large-scale high-speed networks.

Sampling actively collect expected packets instead of dropping unnecessary packets passively due to the limitations of computer resources, thus saving computer resources and reducing burden on the sampled network system. Sampling based data collection is a partial collection technology, which can improve collection efficiency with little devaluation of collection precision [9], [10], [12], [23], [51]. Especially when it comes to detect anomalies and security breaches: a famous Van Jacobson quote reports that ''. . . *If we're keeping per-flow state, we have a scaling problem, and we'll be tracking millions of ants to track a few elephants*'' [27]. We do not need to keep track of all the ants, because the number of elephants in the ants is small. Therefore, sampling is more effective and practical. However, it is inevitable to devaluate collection precision.

There are two kinds of sampling techniques applied in a network system to collect packets. One is integrated sampling technique such as Poisson model. Another is distributed sampling technique, which is more suitable for large-scale high-speed networks and can resist against DDoS attacks.

It is worth noting that sampling size is an important factor, because it affects sampling error, which further decides collection accuracy. For example, sampling error declines with the increasing of sampling size under the same conditions. As shown in Fig. 5, sampling error is inversely proportional to the square root of sample size [10].

Hu *et al.* [9] introduced a simple random sampling technique of statistics for collecting network data and applied it to the IDS. They calculated sample size based on the

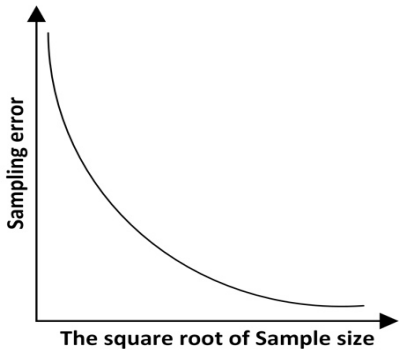


FIGURE 5. Relation between sampling error and sample size.

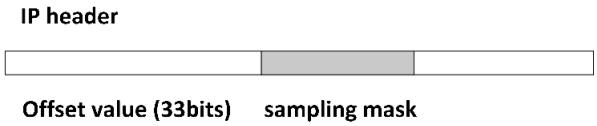


FIGURE 6. A packet sampling model.

sampling theorem that when the sample size is big enough, the distribution of sampling error will be approximately a standard normal distribution. They also made use of inverse sampling to estimate the proportion of packets with attack signature, which they call population attack strength, thereby reducing the sample error. And Zhao *et al.* [10] proposed to use stratified random sampling instead, thus providing a new data collection model for IDS. More specifically, they obtained randomness needed in sampling from IP headers because of following reasons. First, IP header is easy to obtain and does not introduce other data. Second, IP header is usually in plaintext. Third, the identification segment in IP header presents strong randomness, and it is irrelevant to the characteristics of network traffic. Moreover, this segment remains unchanged during network transmission. Fig. 6 illustrates a packet sampling model used in [10]. Because both sampling size and variance in the strata can directly affect the stratified sampling efficiency, they discussed the issue of sample size allocation in strata and presented the scheme for calculating the sample size based on proportional allocation. They found that the proportional allocation is more suitable for stratum that varies a lot than the optimum allocation. For example, experiments showed that stratified sampling based on packet types (e.g., TCP and UDP) [10] can improved data collection efficiency in IDS by only sacrificing precision slightly. Moreover, it can enhance IDS processing performance, especially for the large-scale and high-speed networks. Garcia-Dorado *et al.* [35] proposed a used multi-resolution analysis with wavelets to sample traffic time-series and obtained optimal subsampling levels by comparing the queuing behaviors of subsampled signals with that of original signals at router output. The sample size is adjusted according to the queuing performance impact.

c: TRAFFIC SIMILARITY BASED DATA COLLECTION

Network traffic usually exhibits the characteristic of statistical self-similarity. Wheelus *et al.* [20] proved that although network attacks and anomaly features vary considerably, they share some common features, such as self-similarity, periodicity, repetition and convergence. Because of its statistical self-similarity, Internet traffic is usually modeled with the fractional ARIMA (FARIMA) process instead of the traditional Autoregressive Moving Average (ARMA) process, since it is a behavior model for self-similar time [41], [45], [49]. While FARIMA does improve the performance of traffic prediction for self-similar time series, however, it is time-consuming. Yu *et al.* [45], combined the ARMA and FARIMA processes to model network traffic in order to study their self-similarity.

d: ADAPTIVE DATA COLLECTION

Traditional network data collection method utilizes static strategy that sets static data collecting interval and contents in advance. However, network management practice shows that network data actually correlates with each other. Therefore, by utilizing these correlations NMS does not have to process all the network data. It only needs to dispose some data when others reach a specific condition. Ji *et al.* [11] based on these correlations and variation routines to design a novel Two-Dimensional Adaptive Data Collecting Method (TD-ADCM). Specifically, TD-ADCM selects collecting content according to network context variation and adjusts collecting frequency based on data variation amplitude. The proposed mechanism depends on the analysis of network context and data variation rules.

e: RULE BASED DATA COLLECTION

A rule-based method is brought forward and suggested to combine with existing network technologies in order to collect data among various scenarios (e.g. IDS [52] and honeypot systems [8]). The method encapsulates the logic behind network data collection into business rules, which is organized through algorithms [53]. Rules are used to specify the logics of data collection. For example, in SDN, the controlling rules of Openflow provide basic instructions for flows to forward, change, and drop packets [42], [54]. By studying the characteristics of network systems, Bao and Liu [53] proposed a rule-based method for collecting and processing network data and the business rules are organized through an object-oriented Rete algorithm.

However, the rule-based methods have a disadvantage of high complexity of rules. Moreover, it is difficult to organize rules in the above methods. For example, rules might conflict with each other. As a result, many efforts have been made to solve the conflicts, such as firing the rules by the order of salience, firing the rules by the order of stack or queue, even firing the rules by random. Despite the efforts, none of them can effectively solve the rule conflict problem.

f: LOAD BALANCING BASED DATA COLLECTION

The data collection method based on load balancing is to improve the efficiency of data collection by reducing the burden of large-scale traffic on the collection system by balancing the traffic to multiple network nodes. Balancing network load can help collecting data more easily. For example, Li *et al.* [42] designed a load balancing strategy to ensure equal traffic load distribution among IDS clusters in order to increase IDS performance while still maintaining the network system throughput.

In order to balance network load, Shao *et al.* [55] introduced a new random switching traffic scheduling algorithm based on a data collection tree to alleviate network congestion and collect network data. Their simulation results showed that the proposed scheme significantly reduces the ratio of packet loss and improves the efficiency of data collection system. Moreover, Scap in [30] used a dynamic load balancing mechanism, which applies flow director filters to balance the network traffic load across available queues and cores. It also applies a technique named Receive Side Scaling (RSS) that uses a hash function based on a 5-tuple of packets.

g: FLOW BASED DATA COLLECTION

A flow is a group of packets that share the same source and destination IP address, the same source and destination port, and the same protocol. In other words, flow refers to packets with the same 5-tuple. However, it is called microflow more precisely.

The raw packet collected is a copy of each network frame that circulates in a network system. This bit by bit copy of the network data allows for deep analysis of each network packet. Many IDSs employ a technique named DPI to detect whether the traffic is malicious. This approach has the advantage of detecting hidden attacks in the payload, but large computational overhead can pose a huge burden to large networks. Furthermore, when the payload of the package is encrypted, the problem becomes complex. In this case, it is necessary to evaluate the security of the network system only based on packet header, such as NetFlow and IPFIX [20], [21].

The data collection method based on flow collects flow data [25], [57], which can become more efficient and easier by combining with sampling. In this case, we only need to identify and collect the packets share the same 5-tuple. Further, attacks and anomalies can be detected by processing and analyzing collected flows.

h: STREAM-ORIENTED DATA COLLECTION

Intrusion detection and other network traffic monitoring applications need to collect network traffic beyond the network layer for connection-oriented analysis. For example, we usually need to implement packet reorganization in the application layer in order to analyze business information. Most of the network data collection technologies only provide raw packets, but complex operations, such as flow tracking and TCP stream reassembly, are left to upper applications.

Papadogiannakis *et al.* [30] proposed Scap, a network monitoring framework for stream-oriented traffic collection. Scap provides application-level data collection and reassembled streams by using a kernel module to directly handle flow tracking and TCP stream reassembly.

2) FULL DATA COLLECTION

a: ACTIVE TRAFFIC COLLECTION

Active traffic collection technology appeared to overcome the shortcomings of passive capture. Passive capture could become a hindrance in some circumstances. An example is a collector misses a frame due to some error in the network or at the collector. This existing problem weakens the credibility of collected data. The active traffic collection can be used to collect data through some active technical means (e.g., sending probe packets to a network actively to collect data). Fig. 7 illustrates how the active network data collection works in a measurement system.

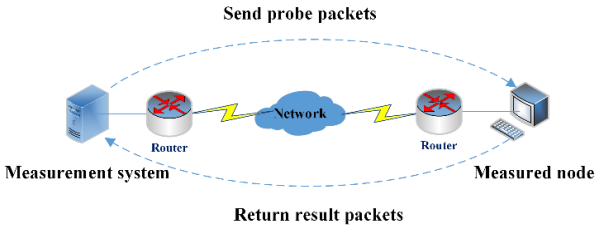


FIGURE 7. An application of active data collection technology.

Slaviero *et al.* [16] proposed an active traffic collection technology to let a collector influence the communication stream under examination. Specifically, they suggested utilizing a TCP retransmission technique based on duplicate acknowledgements to force data resending by a third party. Moreover, Slaviero *et al.* [16] proved its feasibility from both legal and technical perspectives. However, this approach to active traffic capture is limited to force the retransmission of individual TCP frames and is not suitable for collecting other network data.

Actively sending probe packets to a network consumes network bandwidth and burden routers or switches, thus downgrading the network performance. Therefore, active traffic collection needs to be carefully adopted in practical applications.

b: LINEAR SCALING BASED DATA COLLECTION WITH MULTIPLE HARDWARE DEVICES

Network traffic is growing exponentially due to the fast increasing number of the Internet users. As a result, the capability to collect and analyze network traffic needs to scale up accordingly. Using multiple hardware linearly is one of the solutions to scale up data collecting capacity. For example, many existing works tried to achieve linear scaling of network data collection by adding NICs or processors into a server. However, the common network stacks provided by operating systems sacrifice some capabilities in order to

support compatibility, so they are not able to take full advantage of additional hardware for data collection. Paul *et al.* [26] achieved scaled performance of traffic capture up to 40 Gbps with the use of four 10 Gbps NICs by modifying the network stacks. They found that linear scaling can be achieved with multiple NICs in the server.

c: LOCALITY BUFFERING BASED DATA COLLECTION

Papadogiannakis *et al.* [59] proposed a novel approach named Locality Buffering (LB) to improve the performance of network data collection. They found that the application code, data storage structure (e.g., hash table) and attack signatures of network monitoring system all have the locality property of memory access. They enhanced the locality of memory access by reordering the captured packets by grouping together packets with the same application protocol or destination port, and thus improving overall packet collecting and processing performance. An example of data collection technology based on LB is shown in Fig. 8. LB technique rearranged the order of packets, by clustering packets according to the type of packets to enhance locality of memory access. Experimental evaluation showed that using LB can significantly improve the performance of network data collection based on libpcap, e.g., Snort.

FIGURE 8. An example of data collection technology based on LB.

The higher the similarity of packet processing for network data collection applications, the more performance improvement brought by LB. However, there are three main disadvantages of the LB approach. First, the LB approach requires a buffer for reordering packets. Second, the reordering of the packet increases transmission delay and does not apply to services that have high real-time requirements. Third, the LB approach is not universal. Different collection applications need to customize the classification of packets according to their own requirements.

Papadogiannakis *et al.* [60] combined LB and memory mapping to further improve the efficiency of data collection. By mapping a buffer into shared memory, it can reduce the time spent in context switching for delivering packets from kernel to user space. In this way, the performance of network data collection applications can be further improved.

d: DISTRIBUTED DATA COLLECTION

The network system has become increasingly huge and complex with the development of information technology and network equipment. In the meanwhile, network attacks are also becoming more and more complex. As a result, the security monitoring infrastructures need also to grow accordingly. In consequence, a single point in a network system may not be able to diagnose attacks, thus raising a unique set of risks and challenges [30]. Therefore, a distributed system should be deployed to collect network data from multiple network nodes [56]. The data collectors in a distributed data collection system may be software or hardware, which can be selected according to specific requirements [19].

Li and Wu [52] proposed a distributed intrusion detection model to collect data in a distributed manner based on cloud theory. Gad *et al.* [57] used distributed sensors to collect data by also considering that distributed servers are more flexible and safe due to their different locations in the network systems. Moreover, they efficiently partitioned live network data into subsets according to packet header data in order to enable distributed packet capturing. Experimental results in [57] showed that a distributed packet capturing system achieves significantly higher capture rates and efficiency than a single and uncoordinated collector. Chin *et al.* [58] distributed multiple coordinated monitors over a network system, which work with each other to monitor network traffic to detect any potential attacks or anomalies. Fessi *et al.* [7] proposed an architectural distributed IDS that cannot only collect network data, but also analyze the relation between collected events for further processing. IDS with distributed architecture makes it easier to analyze the temporal and causal relationships between security events, resulting in more accurate attack detection. Gad *et al.* [25] proposed a flexible distributed remote packet capture approach with additional self-adaptability and cooperation capabilities. It operates multiple distributed remote packet capturing sensors at arbitrary locations. This method has high scalability, but it requires multiple scattered monitors. Such a distributed structure-based data collection method gives a comprehensive insight into a large computer network. With this method, it is possible to operate multiple distributed remote data collectors in arbitrary locations to capture network data. The cooperation capabilities of the collectors help improving collection performance.

E. FURTHER DISCUSSION, COMPARISON AND ANALYSIS

In this part, we further discuss, compare and analyze a number of data collection methods introduced above based on the objectives and requirements of security-related data collection, as shown in Table 2 and Table 3 with regard to the functional objectives and security objectives, respectively. Note that we only list the papers appearing in Table 2 that consider security functions in Table 3. Other papers that appear in Table 2 but are not listed in Table 3 did not consider security functions at all.

1) PARTIAL DATA COLLECTION

a: TRAFFIC PREDICTION BASED DATA COLLECTION

In [15] and [41], traffic prediction-based data collection methods were introduced to collect network traffic. This kind of methods is widely used for collecting security-related data and has the advantages of high scalability and high flexibility. Moreover, these two traffic collection methods can adaptively adjust collection frequency according to the change of time so as to achieve high collection efficiency without affecting the normal operation of network systems. But hardware is

TABLE 2. Comparison of existing work based on functional objectives.

Tool	Mechanism	Literature	APP	ADA	SCA	STA	GEA	FLB	EFE	N-DES		C
										System	Data	
A Small Number of Monitors	Traffic Prediction	[15]	✓	✓	✓	N/A	N/A	✓	✓	H	H	H
Mobile Terminals		[41]	✓	✓	✓	N/A	✓	✓	✓	L	H	H
IDS	Sampling	[9]	✓	×	×	N/A	✓	✓	✓	H	H	L
IDS		[10]	✓	×	×	N/A	✓	✓	✓	H	H	L
SNMP		[12]	✓	×	✓	✓	✓	×	✓	M	H	H
SNMP		[35]	✓	×	✓	✓	✓	×	✓	M	H	H
Smarter Probes		[27]	✓	×	✓	✓	✓	✓	✓	H	H	M
Distributed Sensor		[25]	×	✓	✓	✓	×	✓	✓	H	H	H
NMS	Adaptive Data Collection	[11]	✓	✓	✓	N/A	✓	✓	✓	H	H	L
N/A	Rule-based Data Collection	[53]	✓	×	✓	×	✓	✓	N/A	N/A	H	L
A Large Number of Sensors	Load Balancing	[55]	✓	✓	✓	N/A	×	×	✓	L	H	H
Scap	Stream-Oriented	[30]	✓	×	✓	N/A	✓	✓	✓	M	M	L
Android Smartphones	NetFlow	[21]	✓	×	✓	✓	✓	✓	N/A	L	H	H
libpcap+tcpdump	Active Traffic Collection	[16]	✓	×	×	N/A	×	N/A	N/A	M	L	L
Inline Taps		[17]	✓	×	✓	✓	✓	×	✓	M	L	H
Multiple Network Interface Cards on Servers	Linear Scaling Technique	[26]	✓	×	✓	✓	✓	✓	✓	N/A	H	H
Multi-core Systems	Multi-core Aware Packet Capture	[40]	✓	×	✓	✓	✓	✓	✓	N/A	H	H
Router+ Collector Server	Port Mirrors	[19]	✓	×	✓	✓	✓	×	✓	L	H	H
Multiple Agents	Distributed Structure	[7]	✓	×	✓	✓	✓	✓	✓	M	H	H
Hardware Probe		[38]	✓	×	✓	✓	×	×	✓	H	H	H
Multiple Sensors		[57]	×	×	✓	N/A	✓	✓	✓	H	H	H
Monitors		[58]	N/A	×	✓	N/A	N/A	✓	✓	M	H	H

Note: L: Low, M: Medium, H: High. ✓: Support this objective, ×: Without consideration, N/A: Not available.

TABLE 3. Comparison of existing work based on security objectives.

Literature	CFD	Verifiability			PP	SPT
		IT	N-RP	AUT		
[27]	✓	×	✓	✓	✓	×
[38]	✓	✓	N/A	N/A	×	×
[21]	✓	✓	✓	✓	✓	×

used as a collection tool in both [15] and [41], so their cost is high. Neither of these two methods of data collection generates useless data. Therefore, non-destructivity can be well supported in terms of data. The method in [15] uses multiple monitors to collect data, which does not destroy the functionality of the network system while the method used in [41] uses mobile terminals to collect data, which affects the normal functions of the mobile terminals. These two works did not consider any security objectives.

b: SAMPLING BASED DATA COLLECTION

In [9] and [10], the statistical sampling algorithms were used to collect network data for software-based IDS. These two methods can support applicability very well. Both of them have low costs and are very flexible to collect network data for intrusion detection. Their collection efficiency is high and does not affect the normal operation of network systems by applying sampling. The above methods can be applied into most networking scenarios, thus they are generic. Both of them does not destroy the functionality of the network system and does not generate useless data. Therefore, non-destructivity can be well supported in terms of both data and system. However, neither of them considers adaptability and scalability. The accuracy of collected data is determined by the sampling algorithm chosen for data collection.

In [12] and [35], SNMP–based network traffic monitoring systems based on statistical sampling algorithms were introduced for network management. The data collection method based on SNMP can be used to collect security-related data, but both methods expend a high cost and are not flexible because of the deployment of MIB and a data analysis server. But also because of the usage of the MIB and the data analysis server, this data collection method is stable, scalable and efficient and can be used in a large-scale network. The data collected by SNMP is real-world because it does not generate useless data. However, SNMP-based data collection

method is not adaptive and has an impact on the operation of the network system.

In [27], Bonelli *et al.* used smart probes to collect data, which is very efficient and stable. This method can be extended and flexibly applied into a variety of data collection scenarios. It does not destroy the functionality of the network system and does not introduce useless data. Therefore, non-destructivity can be well supported in terms of both data and system. But this method's cost is relatively high. Bonelli *et al.* addressed data security and privacy issues by increasing and extending the capabilities of traffic capturing devices, so the smart probes can pre-process the data in the process of data collection in order to solve the security issues of data collection, such as confidentiality and privacy. However, it does not perform integrity check on the data, thus the integrity of the data cannot be guaranteed. Self-protection is not considered in this work.

In [25], Gad *et al.* applied distributed sensors to collect data efficiently with high stability. It can extend its scalability by increasing the number of sensors. But due to the need of multiple hardware sensors, its cost is high. Because distributed sensors can be used in a variety of data collection scenarios, this method holds some additional features such as adaptability and flexibility. The same as [27], this data collection method does not destroy the functionality of the network system and does not introduce useless data. However, this method cannot be used to collect security-related data, so it is not generic.

In short, the advantages of the collection methods based on statistical sampling are that its collection efficiency is high and it does not affect the normal operation of the network system. Most of above works (except [27]) did not consider security objectives in terms of security-related data collection. Thus, they are risky for collecting sensitive and confidential network data for network attack detection and network security measurement.

c: ADAPTIVE DATA COLLECTION

In [11], Ji *et al.* introduced a novel two-dimensional adaptive data collection method for network management. This method ensures that the collected data is accurate by adaptively adjusting collection frequency and collection content. In addition, it also has the advantages with regard to high collection efficiency and low cost. Moreover, this method is generic and scalable and can be flexibly applied into various traffic collection scenarios. This method does not destroy the functionality of the network system and does not generate useless data, so non-destructivity can be well supported in terms of both data and system. All security objectives were not considered in this work.

d: RULE BASED DATA COLLECTION

In [53], Bao and Liu introduced a rule-based method to collect network data by organizing business rules through a Rete algorithm. This method is highly scalable by extending business rules and has low cost. Furthermore, this method is very flexible and can be applied into almost all data collection scenarios. But because of high complexity of rules, it is not stable. The most worrisome is that the efficiency of this method is unknown. This method does not generate useless data, so non-destructivity can be well supported in terms of data. All security objectives were not considered in this work.

e: LOAD BALANCING BASED DATA COLLECTION

In [55], Shao *et al.* introduced a load balancing technology to collect network data. By scheduling communication data throughout a network system, it can improve the collection efficiency without affecting the normal operation of the system. On the other hand, because the scheduling algorithm can balance the communication load adaptively according to the network context, thus this method supports adaptability and scalability. But due to the need to set up a scheduling server and the deployment of the scheduling algorithm, it cannot be flexibly applied into various scenarios and needs to pay a high cost. This method schedules the network communication data of the entire network system. So, any errors of the scheduling algorithm or the scheduling server may destroy the network system. This data collection method does not introduce useless data and thus support non-destructivity well in terms of data. All security objectives were not considered in this work.

f: STREAM-ORIENTED DATA COLLECTION

In [30], Papsadogiannakis *et al.* introduced Scap for data collection. This method has low cost and improves collection efficiency while avoiding affecting the normal operation of the network system by using a subzero packet copy technology and a prioritized packet loss technology. It can be flexibly used in any software-based network traffic collection scenarios. But it lacks adaptability and scalability, thus cannot be applied into complex network contexts. This improved library can be used in most network scenarios. But it may slightly destroy the functionality of the network system and introduce some useless data because of the subzero packet copy technology and the prioritized packet loss technology. All security objectives were not considered in this work.

g: FLOW BASED DATA COLLECTION

In [21], Ariyapala *et al.* used NetFlow technology to collect network data in order to detect malware in a smartphone. This method has the advantages of high stability and scalability. It is generic and can be flexibly applied into various traffic collection scenarios. However, this method requires hardware to support NetFlow technology, thus it has a high implementation cost and is not adaptive. This method does not generate useless data, so non-destructivity can be well supported regarding data. But if there is a lot of network traffic passing the router, it is easy to cause the router's cache insufficient, thus to affect its normal function. Therefore, non-destructivity cannot be supported in terms of system. Furthermore, an anonymous manner was introduced to protect the user's privacy and an encryption algorithm was used to protect the confidentiality of the data. Moreover, data integrity is

guaranteed by message authentication codes and public key signatures. This method also makes use of a group signature protocol to provide anonymity for signers and prevents the system from being destroyed by adjusting the data collected. But self-protection is not considered in this work.

2) FULL DATA COLLECTION

a: ACTIVE DATA COLLECTION

In [16] and [17], active data collection technology was applied to collecting data. Active collection technology can purposefully collect specially needed data, but it need to produce some probe data and thus affects the authenticity of the collected data. Active collection technology may destroy the functionality of the network system, but not so serious. Therefore, non-destructivity cannot be well supported in terms of both data and system. In [16], Slaviero *et al.* used tcpdump to collect network data with low cost, while expensive Inline Taps technology was used in [17]. However, data collection methods based on Inline Taps are more efficient, stable and scalable than data collection methods based on tcpdump. However, the downside of data collection methods based on Inline Taps is not flexible enough. Both two methods lack adaptability and cannot be applied into complex network circumstances. All security objectives were not considered in these two works.

b: LINEAR SCALING BASED DATA COLLECTION WITH MULTIPLE HARDWARE DEVICES

In [26], Paul *et al.* achieved linear scaled performance of traffic capture by modifying the network stacks with the use of multiple NICs. In [40], Jiewu *et al.* introduced a multi-core aware network packet capture module that enables collection to scale with the number of cores. These two data collection methods both have high scalability and stability and can indeed improve the data collection performance by using multiple hardware devices. They are generic and can be flexibly used in a variety of collection scenarios. However, they lack adaptability and are very costly. These two methods do not introduce useless data and thus support non-destructivity well in terms of data. These two works do not satisfy any security objectives.

c: PORT MIRROR BASED DATA COLLECTION

In [19], Han *et al.* used port mirroring technology to collect network data. This method has the advantages of high stability, scalability and high collection efficiency, while its execution cost is high, and it is inflexible and not suitable for a variety of network contexts. It is a generic collection method and can be used to collect security-related data, but cannot adaptively adjust collection strategy. Furthermore, if there is a lot of network traffic passing through a router (or a switch), it will destroy the packet forwarding function of the router (or the switch). But it does not generate useless data. Therefore, non-destructivity can be well supported in terms of data, not for system.

d: DISTRIBUTED DATA COLLECTION

In [7], [38], [57], and [58], data collection methods based on a distributed structure were used to collect data. This kind of methods have high scalability and high collection efficiency. However, the cost of these methods is high since they generally require setting up multiple hardware collection devices. The collection methods proposed in [7], [38], [57], and [58] are not adaptive and cannot adapt to the changing network contexts. These methods do not generate useless data, so non-destructivity can be well supported in terms of data. But multiple agents used in [7] and multiple monitors used in [58] impact the network system's function. In [7], Fessi *et al.* used multiple agents to collect data in a generic, stable and flexible way, while hardware probe used in [38] is not generic and flexible for network data collection. Multiple sensors used in [57] and monitors used in [58] provide a flexible manner for data collection. Furthermore, Gao *et al.* [38] introduced encryption and integrity check technologies to guarantee data confidentiality and integrity. However, other security objectives were not considered in the above type of methods.

From Table 2 and Table 3, we can see that most of the existing work can achieve most of the functional objectives and meet the requirements except that adaptability was not widely considered and supported. However, few of the above reviewed work consider the security objectives. Therefore, the realization of security objectives and requirements becomes a promising research direction in the field of network security-related data collection.

Most of the above existing data collection methods can be used to collect security-related data, but none of them can satisfy all requirements and achieve all objectives. Moreover, the specific characteristics of large-scale heterogeneous networks were seldom considered in the current literature. The existing research only focuses on a single network system architecture. The current literature still lacks generic, comprehensive and extensible description to present network security-related data in the context of heterogeneous networks.

V. OPEN ISSUES AND FUTURE RESEARCH TRENDS

A. OPEN ISSUES

Based on the survey above, we can find a number of open issues in the area of network security-related data collection.

1) HETEROGENEITY

5G era is coming, thus it is critical to implement a proper data collection mechanism in a large-scale heterogeneous network system. However, such a network is composed of many different network systems (e.g., the Internet, WSN, MANET, Internet of Vehicles, Satellite Communications Network, etc.). There are still two unsolved issues in terms of collecting network security-related data. First, it lacks a complete, comprehensive, extendable security-related data description method that can be used in different network contexts. Second, it lacks an efficient data collection

method that can switch adaptively among different network contexts. Therefore, how to design an effective network security-related data collection mechanism for heterogeneous network systems is still an open and practical issue.

2) PRIVACY AND SECURITY

In the future, increasing attention will be paid to the information privacy and security. From Table 3, we can see that most of the existing works can achieve most of the functional objectives and meet the functional requirements, but few of them consider the security objectives. Therefore, ensuring security, protecting privacy and avoiding information leakage during network security-related data collection is an open but significant issue. This problem needs to be solved by studying efficient cryptographic and security techniques.

3) ADAPTABILITY

The data needed to be collected is different (e.g., security-related data) for different application requirements. In order to solve the problem of the validity of data collection, we need to implement specific types of data identification in mass data. Especially in the context of big data and 5G. We observe from Table 2 that there are few existing works achieving the objectives of adaptability. This issue could be solved by efficient data mining and machine learning algorithms or pattern matching techniques.

4) INTELLIGENCE

Table 2 shows that few existing works achieve the objective of intelligence and cannot meet people's requirements on intelligence. How to develop an intelligent and automatic data collection system is still an open issue. The literature expects an efficient machine learning or deep learning algorithm in order to solve this open issue.

5) SCALABILITY WITH LIGHT COMPLEXITY

The rule-based data collection method has high scalability, but it suffers from the rule conflict problem [53]. If we find an efficient solution to solve this problem, scalability can be supported for network security-data collection.

B. FUTURE RESEARCH TRENDS

In addition to the open issues mentioned above, we further propose a number of research directions related to network security-related data collection based on the above survey.

First, intelligent data collection for network security detection and measurement will become a significant research topic. With the development of artificial intelligence, machine learning and deep learning have been used in various fields to achieve intelligent processing and analysis. Data collection should combine with artificial intelligence in the future to achieve intelligent data collection. A system with machine learning capability, which can improve the security, effectiveness and efficiency of data collection, is very desirable. Adaptive sampling technology was received special attention to collect data for a while [11]. However, with the development of hardware, it is not popular in recent years. But with the advances of such new technologies as IoT, big data, and 5G, the literature is highly expecting a technology that can reduce the amount of collected data while keeping the accuracy of data processing result. Therefore, data fusion during data collection or intelligent sampling will become a hot research topic again in the field of data collection.

Second, how to protect data privacy, preserve related user privacy, ensure data security and desensitize data in the process of network security-related data collection is another research hotspot. This issue should also be considered in collecting other types of data for other purposes, not only for security-related data. Specifically, security-related data collection sometimes needs to collect data from users, thus possibly to collect user sensitive information. Some users do not allow others to collect their data due to privacy concern, thus incentive study [61] for security-related data collection could be an interesting research topic.

Third, active data collection technology will become an effective means for network security measurement. This technology can reproduce network system application scenarios and behaviors, thus can accurately obtain network attack and intrusion information. But this kind of data collection mechanisms could mostly burden network equipment and affect the performance of the network system. How to improve the performance of active collection and reduce its impact on the performance of network system will continuously be an important research topic.

Fourth, trust management of security-related data collection is definitely a crucial and significant research topic in order to ensure the veracity of collected data and the trustworthiness of the whole process of data collection, transmission, storage, analysis and processing, as well as usage. Trust management is used to aid automatic decision-making process and plays an important role in the increasing complex network contexts. It helps overcoming perceptions of uncertainty and risk in order to make a decision on a concrete action [65]. However, past studies focused on centralized trust management, which highly depends on a trusted party. This kind of solution is obviously not suitable in the context of a large-scale heterogeneous network system. Novel solutions for distributed trust management is highly expected and should be seriously investigated.

VI. CONCLUSION

Studying network security-related data collection is essential for the detection of network attacks and intrusions, thus contributing to ensure the security of a whole network system. In this paper, we introduced the concept of security-related data collection, specified its requirements and defined its objectives regarding both functionalities and security. Furthermore, we presented a taxonomy and classification of data collection technologies. With regard to data collection technologies, we mainly reviewed data collection nodes, data collection tools and specific data collection mechanisms. We then discussed existing data collection technologies with

regard to the proposed functional and security objectives in order to analyze their pros and cons. Based on our thorough literature survey, we finally indicated a number of open issues and challenges and proposed some future research directions in order to instruct our future research. And hopefully, they can also benefit other researchers and practitioners in this field.

REFERENCES

[1] S. D. Krit and E. Haimoud, "Review on the IT security: Attack and defense," in *Proc. Int. Conf. Eng. MIS*, Agadir, Morocco, 2016, pp. 1–12.

[2] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surv. Tutor.*, vol. 18, no. 2, pp. 1153–1176, Oct. 2016.

[3] F. Sabahi and A. Movaghar, "Intrusion detection: A survey," in *Proc. 3rd Int. Conf. Syst. Netw. Commun.*, Sliema, Malta, 2008, pp. 23–26.

[4] K. H. Ramah, H. Ayari, and F. Kamoun, "Traffic anomaly detection and characterization in the Tunisian National University network," in *Proc. 5th Int. Conf. Res. Netw.*, Coimbra, Portugal, 2006, pp. 136–147.

[5] M. A. Qadeer, M. Zahid, A. Iqbal, and M. R. Siddiqui, "Network traffic analysis and intrusion detection using packet sniffer," in *Proc. 2nd Int. Conf. Commun. Softw. Netw.*, Singapore, 2010, pp. 313–317.

[6] M. Shyu, S. Chen, K. Sarinnapakorn, and L. Chang, "A novel anomaly detection scheme based on principal component classifier," in *Proc. 3rd Int. Conf. Conf. Data Mining Workshop*, Melbourne, FL, USA, 2003, pp. 172–179.

[7] B. A. Fessi, S. BenAbdallah, M. Hamdi, S. Rekhis, and N. Boudriga, "Data collection for information security system," in *Proc. 2nd Int. Conf. Eng. Syst. Manag. Appl.*, Sharjah, United Arab Emirates, 2010, pp. 1–8.

[8] A. N. Singh and R. C. Joshi, "A honeypot system for efficient capture and analysis of network attack traffic," in *Proc. Int. Conf. Signal Process., Commun., Comput. Netw. Technol.*, Thuckafay, India, 2011, pp. 514–519.

[9] L. Hu, K. Zhao, and B. Li, "A data collection model for intrusion detection system based on simple random sampling," in *Computational Methods*, G. R. Liu, V. B. C. Tan, and X. Han, Eds. Dordrecht, The Netherlands: Springer, 2006, pp. 1081–1085.

[10] K. Zhao, M. Zhang, K. Yang, and L. Hu, "Data Collection for Intrusion Detection System Based on Stratified Random Sampling," in *Proc. 4th Int. Conf. Netw., Sens. Control*, London, U.K., 2007, pp. 852–855.

[11] Z. Ji, Z. Kuang, and H. Ni, "A novel two-dimension adaptive data collection method for network management," in *Proc. Int. Conf. Commun. Mobile Comput.*, Yunnan, China, 2009, pp. 237–241.

[12] Y. Ahn and O. Chae, "A design and implementation of network traffic monitoring system for PC-room management," in *Network and Parallel Computing*, H. Jin, G. R. Gao, Z. Xu, and H. Chen, Eds. Berlin, Germany: Springer, 2004, pp. 644–652.

[13] H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, and D. Estrin, "A first look at traffic on smartphones," in *Proc. 10th ACM SIGCOMM Conf. Internet Meas.*, 2010, pp. 281–287.

[14] W. Y. Li, D. Wu, and B. Zhou, "A study of traffic collection techniques for network management and accounting systems," in *Proc. 8th Int. Conf. Comput. Supported Cooper. Work Design*, Xiamen, China, 2004, pp. 578–581.

[15] S. Chen, K. Zeng, and P. Mohapatra, "Efficient data capturing for network forensics in cognitive radio networks," *IEEE Trans. Netw.*, vol. 22, no. 6, pp. 1988–2000, Dec. 2013.

[16] M. Slaviero, A. Granova, and M. Olivier, "Active traffic capture for network forensics," in *Advances in Digital Forensics II*, M. S. Olivier and S. Shenoi, Eds. Boston, MA, USA: Springer, 2006, pp. 215–228.

[17] J. Parry, D. Hunter, K. Radke, and C. Fidge, "A network forensics tool for precise data packet capture and replay in cyber-physical systems," in *Proc. Australas. Comput. Sci. Week Multiconf.*, 2016, pp. 1–10.

[18] J. Malik and R. Kaushal, "CREDROID: Android malware detection by network traffic analysis," in *Proc. 1st ACM Workshop Privacy-Aware Mobile Comput.*, Paderborn, Germany, 2016, pp. 28–36.

[19] H. Han, Z. Chen, Q. Yan, L. Peng, and L. Zhang, "A real-time Android malware detection system based on network traffic analysis," in *Proc. 15th Int. Conf. Algorithms Archit. Parallel Process.*, Zhangjiajie, China, 2015, pp. 504–516.

[20] C. Wheelus, T. M. Khoshgoftaar, R. Zuech, and M. M. Najafabadi, "A session based approach for aggregating network traffic data—The SANTA dataset," in *Proc. 14th Int. Conf. Bioinf. Bioeng.*, Boca Raton, FL, USA, 2014, pp. 369–378.

[21] K. Ariyapala, H. G. Do, H. N. Anh, W. K. Ng, and M. Conti, "A host and network based intrusion detection for Android smartphones," in *Proc. 30th Int. Conf. Adv. Inf. Netw. Appl. Workshop*, Crans-Montana, Switzerland, 2016, pp. 849–854.

[22] G. M. Waku, E. R. Bollis, C. M. F. Rubira, and R. D. S. Torres, "A robust software product line architecture for data collection in Android platform," in *Proc. 9th Brazilian Symp. Softw. Compon., Archit. Reuse.*, Belo Horizonte, Brazil, 2015, pp. 31–39.

[23] A. E. Boualouache, O. Nouali, S. Moussaoui, and A. Derder, "A BLE-based data collection system for IoT," in *Proc. 1st Int. Conf. New Technol. Inf. Commun.*, Mila, Algeria, 2015, pp. 1–5.

[24] B. Sun, F. Yu, K. Wu, and V. C. M. Leung, "Mobility-based anomaly detection in cellular mobile networks," in *Proc. 3rd ACM Workshop Wireless Secur. WiSe*, Philadelphia, PA, USA, 2004, pp. 61–69.

[25] R. Gad, M. Kappes, and I. Medina-Bulo, "Monitoring traffic in computer networks with dynamic distributed remote packet capturing," in *Proc. Int. Conf. Commun. (ICC)*, London, U.K., 2015, pp. 5759–5764.

[26] M. V. V. Paul, R. Bhattacharjee, and R. Rajesh, "Traffic capture beyond 10 Gbps: Linear scaling with multiple network interface cards on commodity servers," in *Proc. Int. Conf. Data Sci. Eng. (ICDSE)*, Kochi, India, 2014, pp. 194–199.

[27] N. Bonelli, A. DiPietro, S. Giordano, G. Procissi, and F. Vitucci, "Towards smarter probes: In-network traffic capturing and processing," in *Trustworthy Internet*, L. Salgarelli, G. Bianchi, and N. Blefari-Melazzi, Eds. Milano, Italy: Springer, 2011, pp. 289–301.

[28] A. Sabiguero, A. Baire, and C. Viho, "Embeding traffic capturing and analysis extensions into TTCN-3 system adaptor," in *Proc. ITG FA 6.2 Workshop Model-Based Test.*, 2006, pp. 1–9.

[29] L. Braun, A. Didebulidze, N. Kammenhuber, and G. Carle, "Comparing and improving current packet capturing solutions based on commodity hardware," in *Proc. 10th ACM SIGCOMM Conf. Internet Meas.*, 2010, pp. 206–217.

[30] A. Papadogiannakis, M. Polychronakis, and E. P. Markatos, "Stream-oriented network traffic capture and analysis for high-speed networks," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 10, pp. 1849–1863, Oct. 2014.

[31] L. Deri and F. Fusco. (Jul. 2017). *Exploiting Commodity Multi-Core Systems for Network Traffic Analysis*. [Online]. Available: http://svn.ntop.org/MulticorePacketCapture.pdf

[32] L. Deri, "nCap: Wire-speed packet capture and transmission," in *Proc. Workshop End-to-End Monitor. Techn. Services*, Washington, DC, USA, 2005, pp. 47–55.

[33] S. Yoon, T. Ha, S. Kim, and H. Lim, "Scalable traffic sampling using centrality measure on software-defined networks," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 43–49, Jul. 2017.

[34] R. S. Panwar and K. M. Sivalingam, "Implementation of wrap around mechanism for system level simulation of LTE cellular networks in NS3," in *Proc. 18th Int. Symp. A World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Macau, China, 2017, pp. 1–9.

[35] J. L. García-Dorado, J. Aracil, J. A. Hernandez, and J. E. L. de Vergara, "A queueing equivalent thresholding method for thinning traffic captures," in *Proc. IEEE/IFIP Netw. Oper. Manag. Symp. Pervasive Manag. Ubiquitous Netw. Service*, Apr. 2008, pp. 176–183.

[36] S. B. Alias, S. Manickam, and M. M. Kadhum, "A study on packet capture mechanisms in real time network traffic," in *Proc. Int. Conf. Adv. Comput. Sci. Appl. Technol.*, Kuching, Malaysia, 2013, pp. 456–460.

[37] F. Schneider, J. Wallerich, and A. Feldmann, "Packet capture in 10-gigabit ethernet environments using contemporary commodity hardware," in *Passive and Active Network Measurement*, S. Uhlig, K. Papagiannaki, and O. Bonaventure, Eds. Berlin, Germany: Springer, 2007, pp. 207–217.

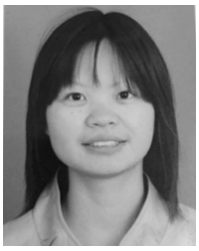
[38] K. Gao, J. Liu, J. Guo, and R. An, "Study on data acquisition solution of network security monitoring system," in *Proc. Int. Conf. Inf. Theory Inf. Security*, Beijing, China, 2010, pp. 674–677.

[39] L. Zabala, A. Ferro, and A. Pineda, "Modelling packet capturing in a traffic monitoring system based on Linux," in *Proc. Perform. Eval. Comput. Telecommun. Syst.*, Genoa, Italy, 2012, pp. 1–6.
[40] F. Fusco and L. Deri, "High speed network traffic analysis with commodity multi-core systems," in *Proc. 10th ACM SIGCOMM Conf. Internet Meas.*, 2010, pp. 218–224.
[41] W. Jiewu, F. Wentao, H. Chunjing, and Z. Xing, "User traffic collection and prediction in cellular networks: Architecture, platform and case study," in *Proc. 4th Int. Conf. Netw. Infrastruct. Digit. Content*, Beijing, China, 2014, pp. 414–419.
[42] B. Li, P. Liu, and L. Lin, "A cluster-based intrusion detection framework for monitoring the traffic of cloud environments," in *Proc. 3rd Int. Conf. Cyber Secur. Cloud Comput. (CSCloud)*, Beijing, China, 2016, pp. 42–45.
[43] J. Song, J.-W. Choi, and S.-S. Choi, "A malware collection and analysis framework based on darknet traffic," in *Neural Information Processing*, T. Huang, Z. Zeng, C. Li, and C. S. Leung, Eds. Berlin, Germany: Springer, 2012, pp. 624–631.
[44] Q. Wen, Z. Zhao, R. Li, and H. Zhang, "Spatial-temporal compressed sensing based traffic prediction in cellular networks," in *Proc. 1st Int. Conf. Commun. China Workshops (ICCC)*, Bejing, China, 2012, pp. 119–124.
[45] Y. Yu, M. Song, Y. Fu, and J. Song, "Traffic prediction in 3G mobile networks based on multifractal exploration," *Tsinghua Sci. Technol.*, vol. 18, no. 4, pp. 398–405, Aug. 2013.
[46] X. Wang and X. Shan, "A wavelet-based method to predict Internet traffic," in *Proc. Int. Conf. Commun., Circuits Syst. West Sino Expo.*, Chengdu, China, 2002, pp. 690–694.
[47] B. Krithikaivasan, K. Deka, and D. Medhi, "Adaptive bandwidth provisioning envelope based on discrete temporal network measurements," in *Proc. IEEE INFOCOM*, Hong Kong, Mar. 2004, pp. 1786–1796.
[48] A. Khotanzad and N. Sadek, "Multi-scale high-speed network traffic prediction using combination of neural networks," in *Proc. Int. Joint Conf. Neural Netw.*, Portland, OR, USA, 2003, pp. 1071–1075.
[49] A. D. Doulamis, N. D. Doulamis, and S. D. Kollias, "An adaptable neural-network model for recursive nonlinear traffic prediction and modeling of MPEG video sources," *IEEE Trans. Neural Netw.*, vol. 14, no. 1, pp. 150–166, Jan. 2003.
[50] H. Feng, Y. Shu, S. Wang, and M. Ma, "SVM-based models for predicting WLAN traffic," in *Proc. Int. Conf. Commun.*, Istanbul, Turkey, 2006, pp. 597–602.
[51] M. Canini, D. Fay, D. J. Miller, A. W. Moore, and R. Bolla, "Per flow packet sampling for high-speed network monitoring," in *Proc. 1st Int. Commun. Syst. Netw. Workshops*, Bangalore, India, 2009, pp. 1–10.
[52] H. Li and Q. Wu, "A distributed intrusion detection model based on cloud theory," in *Proc. 2nd Int. Conf. Cloud Comput. Intell. Syst.*, Hangzhou, China, 2012, pp. 435–439.
[53] T. Bao and S. Liu, "A method for network data collection and processing in the pervasive computing environment," in *Proc. 1st Int. Symp. Pervasive Comput. Appl.*, Urumqi, China, 2006, pp. 599–603.
[54] S. Dotcenko, A. Vladyko, and I. Letenko, "A fuzzy logic-based information security management for software-defined networks," in *Proc. 16th Int. Conf. Adv. Commun. Technol.*, Pyeongchang, South Korea, 2014, pp. 167–171.
[55] S. Shao, S. Guo, X. Qiu, L. Meng, and C. Zhong, "A random switching traffic scheduling algorithm for data collection in wireless mesh network," in *Proc. 16th Asia–Pacific Netw. Oper. Manage. Symp.*, Hsinchu, Taiwan, 2014, pp. 1–4.
[56] J. Liu, H. Wen, X. Hou, G. Sun, and S. Zhu, "Traffic collection and analysis system," in *Social Computing*, W. Che *et al.* Eds. Singapore: Springer, 2016, pp. 229–234.
[57] R. Gad, M. Kappes, R. Mueller-Bady, and I. Medina-Bulo, "Header field based partitioning of network traffic for distributed packet capturing and processing," in *Proc. 28th Int. Conf. Adv. Inf. Netw. Appl.*, 2014, pp. 866–874.
[58] T. Chin, X. Mountrouidou, X. Li, and K. Xiong, "An SDN-supported collaborative approach for DDoS flooding detection and containment," in *Proc. MILCOM*, Tampa, FL, USA, 2015, pp. 659–664.
[59] A. Papadogiannakis, D. Antoniades, M. Polychronakis, and E. P. Markatos, "Improving the performance of passive network monitoring applications using locality buffering," in *Proc. 15th Int. Symp. Modeling, Anal., Simulation Comput. Telecommun. Syst.*, Istanbul, Turkey, 2007, pp. 151–157.
[60] A. Papadogiannakis, G. Vasiliadis, D. Antoniades, M. Polychronakis, and E. P. Markatos, "Improving the performance of passive network monitoring applications with memory locality enhancements," *Comput. Commun.*, vol. 35, no. 1, pp. 129–140, Jan. 2012.
[61] X. Ji, D. Zhao, H. Yang, and L. Liu, "Exploring diversified incentive strategies for long-term participatory sensing data collections," in *Proc. 3rd Int. Conf. Big Data Comput. Commun. (BIGCOM)*, Chengdu, China, 2017, pp. 15–22.
[62] L. F. Zhang, Z. Yan, and R. Kantola, "Privacy-preserving trust management for unwanted traffic control," *Future Generat. Comput. Syst.*, vol. 72, pp. 305–318, Jul. 2017.
[63] L. M. He, Z. Yan, and M. Atiquzzaman, "LTE/LTE—A network security data collection and analysis for security measurement: A survey," *IEEE Access*, vol. 6, pp. 4220–4242 2018.
[64] G. Liu, Z. Yan, and A. W. Pedryczc, "Data collection for attack detection and security measurement in mobile ad hoc networks: A survey," *J. Netw. Comput. Appl.*, vol. 105, pp. 105–122, Mar. 2018.
[65] Z. Yan, P. Zhang, and A. V. Vasilakos, "A survey on trust management for Internet of Things," *J. Netw. Comput. Appl.*, vol. 42, pp. 120–134, Jun. 2014.
[66] P. Woods. (Jul. 2017). *Libpcap MMAP Mode on Linux*. [Online]. Available: http://public.lanl.gov/cpw/
[67] P. Yan and Z. Yan, "A survey on dynamic mobile malware detection," *Software Quality J.*, pp. 1–29, May 2017, doi: 10.1007/s11219-017-9368-4.
[68] L. Chen, Z. Yan, W. D. Zhang, and R. Kantola, "TruSMS: A trustworthy SMS spam control system based on trust management," *Future Generat. Comput. Syst.*, vol. 49, pp. 77–93, Aug. 2015.
[69] Y. Shen, Z. Yan, and R. Kantola, "Analysis on the acceptance of global trust management for unwanted traffic control based on game theory," *Comput. Security*, vol. 47, pp. 3–25, Nov. 2014.
[70] Z. Yan, R. Kantola, and Y. Shen, "A generic solution for unwanted traffic control through trust management," *New Rev. Hypermedia Multimedia*, vol. 20, no. 1, pp. 25–51, Oct. 2013.

HUAQING LIN received the B.Sc. degree in information security from Guizhou University, Guiyang, China, in 2016. He is currently pursuing the master's degree with the State Key Laboratory on Integrated Services Networks, Xidian University. His research interests are in data collection, machine learning, mobile security, and trust management.

ZHENG YAN (M'06–SM'14) received the B.Eng. degree in electrical engineering and the M.Eng. degree in computer science and engineering from the Xi'an Jiaotong University, Xi'an, China, in 1994 and 1997, respectively, the second M.Eng. degree in information security from the National University of Singapore, Singapore, in 2000, and the Licentiate of science and the D.Sc. degree in technology in electrical engineering from the Helsinki University of Technology, Espoo, Finland. She is currently a Professor with the Xidian University, Xi'an, China, and a Visiting Professor with the Aalto University, Espoo, Finland. Her research interests are in trust, security, and privacy, social networking, cloud computing, networking systems, and data mining. She serves as an organization and program committee member for over 80 international conferences and workshops. She is also an Associate Editor of many reputable journals, e.g., the IEEE INTERNET OF THINGS Journal, the *Information Sciences*, the *Information Fusion*, JNCA, the IEEE ACCESS, and SCN.

YU CHEN received B.Eng. degree in information security from the Huazhong University of Science and Technology in China, the M.Sc. degree in security and mobile computing from Aalto University and the second M.Sc. degree in security and mobile computing from the Norwegian University of Science and Technology in 2010, and the D.Sc. degree in communication and computer sciences from École Polytechnique Fédérale de Lausanne (EPFL), Switzerland, in 2015. She was a Research Assistant with HCI Group, EPFL from 2010 to 2014 and a Research Intern with the Nokia Research Center, Helsinki, in 2010. She is currently a Post-Doctoral Researcher with the Department of Informatics, University of California at Irvine, Irvine, CA, USA. She was a recipient of the NordSecMob Erasmus Mundus Scholarship in 2008 and the Swiss National Science Foundation Early Postdoc Mobility Fellowship in 2015.



LIFANG ZHANG received the B.S. degree in electrical engineering from Beijing Forestry University, Beijing, China, in 2012, and the M.S. degree in communication engineering from Aalto University, Espoo, Finland, in 2016. She is currently a Research Assistant with Aalto University.