Detecting and Preventing Cyber Insider Threats: A Survey

Liu Liu, Olivier De Vel, Qing-Long Han, *Senior Member, IEEE*, Jun Zhang, and Yang Xiang, *Senior Member, IEEE*

***Abstract*—Information communications technology systems are facing an increasing number of cyber security threats, the majority of which are originated by insiders. As insiders reside behind the enterprise-level security defence mechanisms and often have privileged access to the network, detecting and preventing insider threats is a complex and challenging problem. In fact, many schemes and systems have been proposed to address insider threats from different perspectives, such as intent, type of threat, or available audit data source. This survey attempts to line up these works together with only three most common types of insider namely traitor, masquerader, and unintentional perpetrator, while reviewing the countermeasures from a data analytics perspective. Uniquely, this survey takes into account the early stage threats which may lead to a malicious insider rising up. When direct and indirect threats are put on the same page, all the relevant works can be categorised as host, network, or contextual data-based according to audit data source and each work is reviewed for its capability against insider threats, how the information is extracted from the engaged data sources, and what the decision-making algorithm is. The works are also compared and contrasted. Finally, some issues are raised based on the observations from the reviewed works and new research gaps and challenges identified.**

***Index Terms*—Insider threats, data analytics, machine learning, cyber security.**

I. Introduction

THE DAILY operations of governments, enterprises and organisations rely on networked infrastructures that interconnect computers and related devices across departments and networks to facilitate data accessibility and sharing of computer resources. Protecting such infrastructures from various cyber attacks and threats is of paramount importance [1]. According to the Clearswift Insider Threat Index (CITI) annual report 2015 [2], 92% respondents (organisations) claimed that they have experienced IT or data security incidents in the past 12 months and 74% of these breaches were originated by insiders. Thus addressing threats posed by insiders is the top priority for achieving full protection of networked infrastructures.

Despite almost two decades of research seeking ways to detect and prevent insider threats, the advancement of modern networks has quickly outpaced these efforts. As a result, victims continue to report huge losses because of malicious insiders. This may be due to one or more of the following reasons: 1) the existing solutions do not pay enough attention on the early indications of an arising malicious insider, most of which do not raise alerts until damaging behaviours have occurred; 2) most of the solutions rely only on an individual audit data source, diminishing insights into the threats; and 3) conventional data analytics counts too much on domain knowledge in extracting features or establishing rules, resulting in a limited capability against evolving threats. Therefore, this survey collates the most up-to-date representative schemes and systems, in an attempt to explore the full trace left by a malicious insider, highlight the pros and cons of the established works, and suggest a research roadmap that may direct us to a better solution.

In the latest CERT Coordination Centre (CERT/CC) technical report [3], an insider threat is defined as a malicious insider who intentionally exploits his or her privileged access to the organisation's network, system and data, taking actions that negatively affect the confidentiality, integrity or availability of the organisation's information and ICT infrastructures. These insiders were defined as "traitors" in an earlier survey [4], while the other major type of questionable insiders were defined as being impersonated by "masqueraders" to pose a threat to the organisation [4]. In addition, an insider threat may also be posed by a legitimate user unintentionally making a mistake (i.e., "unintentional perpetrator") [5]. Since a masquerader often penetrates the ICT system using stolen credentials or a compromised computer that belongs to a legitimate user, in terms of intent, only traitors themselves are explicitly malicious. However, it has been well justified that no matter whether it is deliberate or not, malicious or unusual behaviour will deviate from normal behavioural patterns [4]. Thus, this survey focuses on all the threats related to the above-mentioned types of insider without distinguishing the intents.

According to the literature, the most commonly seen insider threats are 1) data exfiltration, 2) violations against data integrity or availability and 3) sabotage of ICT systems [5], [6]. Technically, traitors and unintentional perpetrators are able to fulfil these threats straightway. A masquerader may pose the same threats via an intrusion campaign that consists of social engineering, eavesdropping

Manuscript received March 29, 2017; revised November 26, 2017; accepted January 12, 2018. Date of publication February 1, 2018; date of current version May 22, 2018. *(Corresponding author: Yang Xiang.)*

L. Liu, Q.-L. Han, J. Zhang, and Y. Xiang are with the School of Software and Electrical Engineering, Swinburne University of Technology, Hawthorn, VIC 3122, Australia (e-mail: liuliu115621@gmail.com; qhan@swin.edu.au; junzhang@swin.edu.au; yxiang@swin.edu.au).

O. De Vel is with the Defence Science and Technology Group, Department of Defence, Edinburgh, SA 5111, Australia (e-mail: olivier.devel@dst.defence.gov.au).

Digital Object Identifier 10.1109/COMST.2018.2800740

TABLE I
TYPES OF INSIDER AND RELEVANT THREATS

Type of insider	Kill chain	Threat
Masquerader	Reconnaissance	Port scan Network vulnerability scan Web application vulnerability scan database vulnerability scan
	Weaponisation	Social engineering
	Delivery	Email spam (URL or attachments) Malicious or phishing websites Removable media
	Exploit	Privilege escalation
	Install	RAT or backdoor
	C2	DDoS Email spam Click fraud and bitcoin mining
	Actions on objectives	Data exfiltration Violation against data integrity or availability Sabotage of ICT systems
Traitor		
Unintentional perpetrator		

and packet sniffing, malware delivery and installation, and etc. [4]. Since the advanced persistent threat (APT) intrusion kill chain [7] represents a latest intrusion campaign paradigm, it is employed in this survey as a taxonomy to accommodate the early-stage threats posed during the incubation period of a masquerader. In particular, an intrusion kill chain is defined as a systematic process to target and engage an attacker by creating desired effects which, in the context of APT, involves seven phases, namely reconnaissance, weaponisation, delivery, exploitation, installation, command and control (C2) and actions on objectives [7]. Such a kill chain actually enables this survey to look at typical insider threats and the early-stage threats on the same page. We propose that if any of these threats can be addressed, it has successfully prevented an insider who is committing a violation and thus, we regard the two categories both as insider threats without distinction in the following text. Table I summarises the types of insider and all the relevant threats.

Audit data source plays a significant role in determining the capability, effectiveness and efficiency of a proposed scheme. Without an appropriate audit data source, one can not expect a valid outcome no matter what analytical technique is adopted. As such, this survey reviews the countermeasures in terms of the engaged data sources. Host-based and network-based are traditionally the most popular two categories of data source in designing an anomaly-based intrusion detection system (IDS) [8]. Typical host-based data sources include system calls, Unix shell commands, keyboard and mouse dynamics, and various host logs to which a behavioural analysis of programs, users or hosts is applicable. Network traffic and logs are the most common examples of a network-based data source, from which information can be extracted to modelling the networking behaviours of any users [9], hosts, IP addresses, TCP flows and so forth. Except for the above two categories, contextual data sources are considered as the third category in this survey, which are meant to provide contextual information such as the human resources (HR) and psychological data in regard of a human user. Contextual data sources have shown great potential in conducting intent analysis and validating the suspicious behaviours reported by a conventional analytics [10], [11]. In line with the three categories of data source, this survey will look into a large number of the existing schemes and systems, particularly concerning the capability against insider threats, how the features or useful information are extracted from the employed data sources, the modelling approach and the specific decision-making algorithm. Furthermore, for each category of data source, we give a short review comparing and contrasting the relevant works from a technical perspective such as the pros and cons of the modelling approaches and decision-making algorithms used.

Salem's survey [4] might be the earliest one that focuses specifically on detecting insider attacks. In that survey, malicious insiders are categorised into traitors and masqueraders, and some insider threats are loosely exemplified without a systematic view. Moreover, its most pages are spent in reviewing the schemes regarding host-based user profiling. Hence, in terms of both depth and breadth, it can only be regarded as a partial reference for the state-of-the-art research of detecting and preventing insider threats. CERT's technical reports [3], [6] deeply investigate the indicators an insider threat may have expressed and suggest a series of practices which are essential for mitigating insider threats. They are useful guides and handbooks for organisational decision makers but do not really fill the gap with sufficient academic materials derived. Therefore, we are motivated to complete a new survey aiming to provide more comprehensive insights into this research from a technical perspective. Particularly, this survey makes the following contributions:

- It identifies three types of insider and deals with each type in the same way by ignoring intent.
- Conventional insider threats and some other relevant early-stage threats are put on the same page using the APT intrusion kill chain.

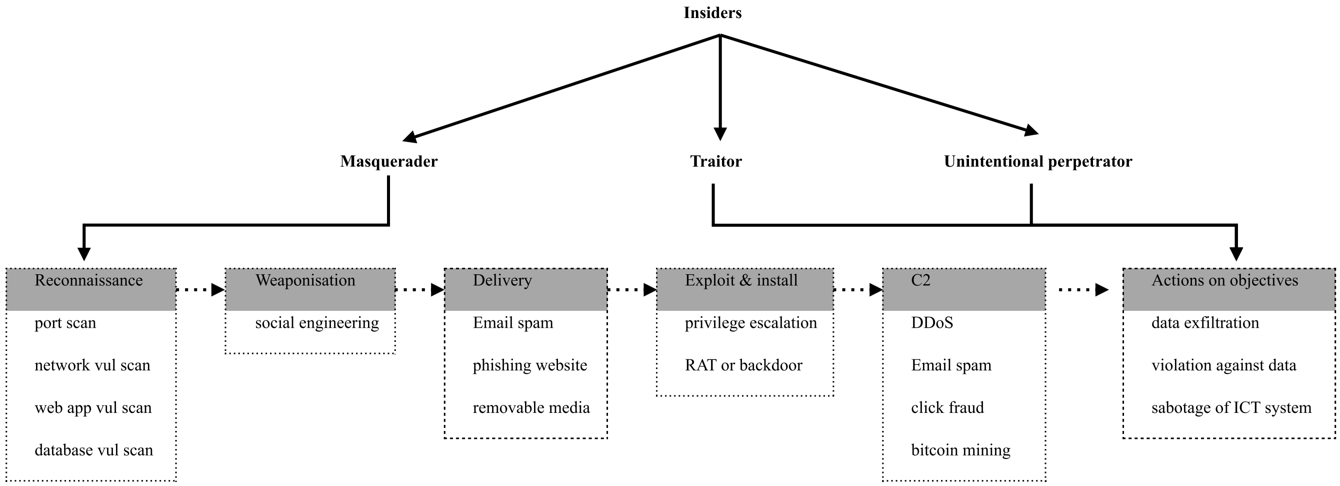


Fig. 1. Taxonomy of the Insider Types and Specific Insider Threats.

- A large number of recent works are organised in terms of data source, each of which is presented particularly from a data analytics perspective,
- Contextual data-based analytics is recognised as a promising new technical category, with some representative works reviewed.
- Some critical research challenges are identified.

This survey is organised as follows. Section II introduces the insider threats according to the types of insider. Sections III to V detail host, network and contextual data-based analytics respectively, each of which also comes with a brief introduction to the common data sources and a technical summary. Section VI discusses some observations from the reviewed works and identifies some important research challenges. Finally, Section VII concludes this work.

II. Types of Insider and Relevant Threats

In this section, we introduce the types of insider and what threats they can pose, by gathering and recompiling the materials obtained from the literature. The APT intrusion kill chain is employed to arrange all the threats from early to late stages. In the meantime, this section provides the details about vulnerabilities of current systems to the threats, typical attack vectors and how a victim can be exploited. Figure 1 illustrates the types of insider and relevant threats.

We assume that in most cases, a masquerader is an outside attacker penetrating the system via the kill chain. A penetration normally begins with 'reconnaissance', during which the attacker is gathering the targeted victim's information by various means for use in a future attack [12]. Essentially, this phase is aimed at exploring the victim environment's computer systems, networks and applications for vulnerabilities [13], which often relies on the following means, i.e., port scanning [14], network vulnerability scanning [15], Web application vulnerability scanning [16], database vulnerability scanning [17] and etc. The second phase 'weaponisation' is conducted quietly on the attacker side without any interaction with the victim environment. By leveraging the knowledge learnt from the former phase and social engineering [18], the attacker creates a weaponised deliverable coupled with malicious payloads such as remote access trojans (RAT) [19], rootkit backdoors [20] and keyloggers [21]. However, at this time, the attacker still remains outside the victim environment, leaving no footprints in the audit data. As this survey focuses primarily on dealing with insider threats from a data analytics perspective, we ignore the early two phases.

The attacker really start engaging the victim environment by launching the 'delivery' phase. Social engineering based phishing is the most common means that delivers the malware into the victim's host; for example, the victim may receive an unsolicited email with URLs redirecting to malicious Web sites and/or attachments such as executable binary, PDF and MS Office document coupled with malicious payload [22], [23]. Alternatively, the victim may be directly tempted to visit a malicious website when surfing the Internet. No matter how it is being deceived, once landing to a malicious website, the malware will be silently delivered to the victim's host [24], [25]. Removable media such as USB thumb drives and USB mass storage devices is a relatively clumsy means to deliver the malware but still pervasively dangerous [26], [27], which usually infects the victim's host through a featured malicious 'autorun' payload [28].

The weapon delivered to the victim's host will be self-executed and, by exploiting the vulnerabilities of the operating system and/or the applications, it installs the malware to keep a door open for the attacker. These actions correspond to the two phases 'exploitation' and 'installation'. To be more specific, an exploit is a sequence of operations varying according to underlying hardware, operating systems and applications but its fundamental objective is always the same: gaining control over the victim's host through escalation of privilege and then attempting to steal credentials and install the RAT or backdoor. A general example is that an exploitation takes advantage of a certain known or unknown (zero-day [29]) vulnerability to launch a buffer overflow attack. With the escalated privilege, the attacker can choose to act on the victim environment immediately such as accessing or damaging sensitive data [30], [31]) or, quietly install and propagate the malware [32].

In order to maintain the threat persistently, the attacker keeps communicating with the compromised host (i.e., bot) via a C2 channel [33]. The C2 channel is established by the installed RAT or backdoor, which allows the attacker to instruct the bot to perform some desired operations. When there are a number of hosts turned into bots, they can be organised as a botnet in an ad-hoc manner, operating in either a Client-server (CS) or a peer-to-peer (P2P) model. A CS model based botnet can frequently change its server's IP address by subscribing to a dynamic Domain Name System (DDNS) service to avoid detection [34], [35]. In contrast, without relying on a centralised server, a P2P model based botnet is harder to be detected and eliminated, since in such a network each node acts as bot master and bot client at the same time and it can still work properly even if some of the nodes have been taken down [36], [37]. Some destructive attacks may be implemented straightforwardly through a botnet, such as Distributed Denial-of-Service (DDoS) that disrupts the victims' network by incoming traffic flooding [38], [39] and email spam that dumps a numerous number of unwanted, advertising or malicious emails into the victim environment [40]. Alternatively, an attacker may exploit the bots' resources for running a profitable business, such as click fraud [41] and Bitcoin mining [42].

Then, we look at the threats that arise when an outside attacker has morphed into a malicious insider. The last phase is referred to as 'actions on objectives' in the kill chain, indicating that the masquerader has accomplished the first six phases and now can take actions to achieve the original objectives [7]. Since traitors and unintentional perpetrators reside internally, they can pose the same threats without having to walk through the intrusion campaign. Generally, data exfiltration is the most dangerous threat, which can be formally defined as "unauthorised copying, transferring, or retrieving of data from a computer or server" [3]. In practice, its actions exist in various forms such as 1) unauthorised access to or use of corporate information, 2) unintentional exposure of private or sensitive data and 3) theft of intellectual property (IP) [6]. Sometimes, traitors and unintentional perpetrators are allowed to take these bad actions immediately; for example, they can browse, search, download and print documents which they are not authorised to access or do not need to access, and transfer protected data outside using removable media [43], cloud boxes [44], [45] or email attachments [31], [43]. In contrast, due to the nature of remote control, masqueraders may have to operate in a more sophisticated manner; for example, they may collect only critical information about the compromised host and the victim's personal activities, e.g., operating system version, enabled ports, password and credit card information [46], which then can be sold as information asset or used as a bridge for lateral movement [7], [31], [32]. Violation of data integrity or availability is the second major threat posed to a victim environment. The definition of such a threat can be described as "disappearance or damage in which a correct data copy is no longer available to the organisation" [47]. Known actions relating to this threat include: 1) changing file extensions to confuse users, 2) enciphering/deciphering sensitive data [31], and 3) tampering with files [43]. Ransomeware [48] is a more realistic example, which penetrates into a victim host, encrypts the data and demands payment for decryption. This is a typical violation of data availability. The attacker is also able to pose a threat of sabotage of the ICT system, which is defined as "an insider's use of ICT to direct specific harm at an organisation or an individual [6]". For example, as previously mentioned, the masquerader can operate the P2P model based botnet to launch a DDoS attack from the inside against the victim ICT system [38], resulting in a much more serious consequence than ordinary DoS attacks [49].

III. Host-Based Analytics

In this survey, as shown in Figure 2, there are three categories of data source being taken into consideration: 1) host, 2) network, and 3) contextual. This section will focus on host-based analytics.

Host-based analytics are working with data collected from each individual host (computer), ranging from operating system low-level data such as system calls to application-level data such as shell command lines, keystroke/mouse dynamics, *nix syslog, Windows event log and etc. These data sources are able to reflect how a host behaves and the human user's interactive behaviour with the host. As such, they can find wide applications in addressing insider threats.

A system call is made when a computer program requests a service from the kernel of the operating system where these services are primarily provided to manage and access a computer's hardware or kernel-level resources such as CPU, memory, storage, network and process [50]. There are many different ways to collect system calls, such as the *nix operating systems' built-in Auditing System (*auditd* daemon) or *strace*/*ptrace* programs. By working on different programs, the captured system calls can be analysed to identify a broad range of cyber security threats. For example, analysis of how a privileged process is being executed has been proven effective against intrusions [51], [52] and malware [31]. Additionally, investigation of system calls that are specific to file operation is capable of uncovering a malicious insider who is attempting to access protected data [53].

Commands and keystroke/mouse dynamics are more concerned with how a user is operating the host. The commands usually exist in the form of a sequence while the keystroke/mouse dynamics need a specialised model to characterise their features [54]. Collection of a user's input data may not be easy due to privacy concerns, relying on whether the built-in command logger [55] or a third-party command/keystroke/mouse recorder [56], [57] is available on the targeted host. Since this category of data sources contain information that enables identifying genuine users from a behavioural biometrics' perspective, they are best suited for detecting masqueraders.

Operating system's built-in logging capability can be leveraged to record a variety of system events such as authentication, system daemon catch-all, kernel messages, process, policy change and etc. According to the type of operating system, in practice they are called *nix syslog and Windows event log respectively. Although sometimes such a log is

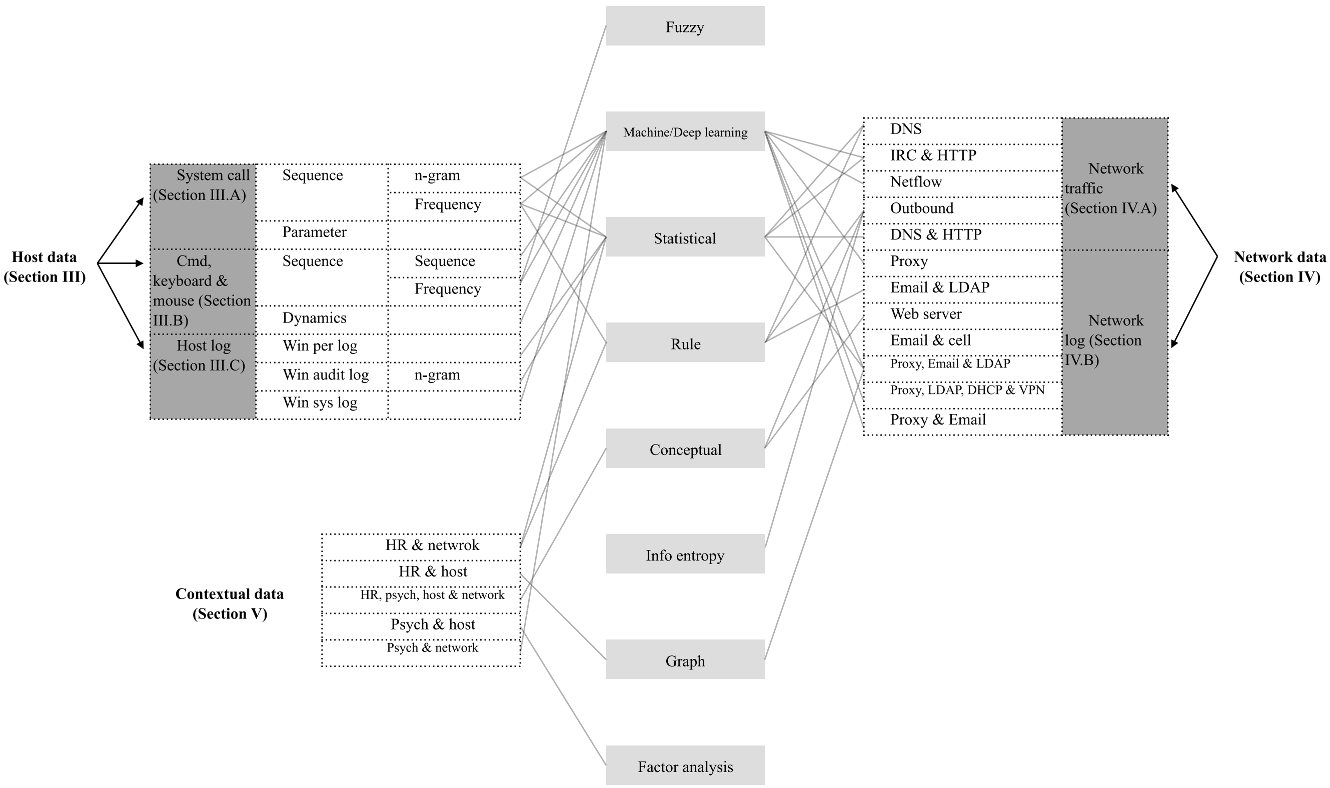


Fig. 2. Data sources and analytics.

verbose due to a large number of duplicated entries, the information contained may still be worth examining. For example, an unusually long sequence of authentication failures may indicate a brute-force password attack. However, due to the extreme redundancy and complexity, raw logs are indeed not very effective to reflect indications of compromise. One possible solution is to spend some effort in processing raw logs and extracting features for addressing a certain type of cyber security threat. Instead, it is also possible to reengineer the logging capability to collect only the information specific to the targeted threats; for example, aiming at malicious insiders, the RUU ("are you you?") logger is tailored to merely look after process, registry and file operation-related system events [58].

A. System Calls

System call is the channel that a program communicates with the kernel of the operating system. It has been extensively analysed for detecting intrusions, malware and insider threats. The following example represents a subsequence of system call captured from a *sendmail* program.

```
open, read, remap, remap, open,
getrlimit, remap, close
```

Most of the existing schemes are working on such sequences of system call, while a few have looked deeper into the parameters of a system call. In regard of sequences of system call, there are two common technical categories: 1) sequence-based that constructs n-grams from a sequence and applies different models to analyse the n-grams and, 2) frequency-based that transforms the sequences into equal-length frequency vectors to which, then, statistical or machine learning algorithms are applied. Apart from statistical and machine learning algorithms, sequence, rule-based, graph-based and deep learning algorithms also have found applications in dealing with system calls. The following paragraphs will introduce some representative schemes in detail.

Analysis of system call is pioneered by Forrest and Hofmeyr in the 1990s [51], [69]. For the purpose of intrusion detection, the early schemes such as the 'lookahead' [51] and its refined variant [52] focus primarily on the sequences of system call that reflect a certain privileged process (e.g., *sendmail* and *ftp*). When a baseline database (i.e., a set of n-grams) is built using a fixed size window (i.e., *n*) sliding across a set of normal sequences, the n-grams obtained from a testing sequence are compared with the database contents, revealing an anomaly if the percentage of mismatch is beyond a defined threshold. Considering that a malicious payload only impacts on a small section of the sequence, the refined 'lookahead' simply counts mismatches within a suspicious subset of the testing sequence, leading to a lower computational complexity. Since the Markov chains and Hidden Markov Models (HMMs) are able to characterise the sequence of state (system call) transitions more accurately, they are better alternatives to the early schemes [59], [60], in which cases the joint probabilities of the n-grams are usually employed as a metric of abnormality. Artificial neural network (ANN)-based schemes [61], [62]

TABLE II
TAXONOMY OF SYSTEM CALL BASED ANALYTICS

Threat type	Model	Tech category	Algorithm	Data source
Intrusion & malware	n-gram	statistical	sequence match [51] [52]	System call sequence
	n-gram	statistical	Markov [59] [60]	
	n-gram	machine learning	feedforward neural network [61] [62]	
	n-gram	deep learning	recurrent neural network (RNN) [63]	
	frequency	statistical	LLRT, LR [64]	
	frequency	machine learning	kNN [65] [66]	
		machine learning	kMC [67]	
		machine learning	SVM [68] [64]	
Insider threats		rule	signature match [30]	System call parameter
		graph	minimum description length (MDL) [53]	

are even more accurate because of ANN's strong ability to discover non-linear correlations between inputs (n-grams) and outputs (normal or anomalous). Recently, deep learning (ANN's rebranded descendant)-based scheme has also came out, which applies a higher number of diverse hidden layers to extract features from the n-grams [63]. However, neural network family's advantage in accuracy comes with a largely increased computational cost and its effectiveness has not been proven for unsupervised learning which is a more common case in practice.

Apart from the n-gram-based schemes, the other schemes tend to analyse a system call based on its frequency. In particular, a sequence of system call is transformed into a fixed-length frequency vector according to the occurrence number of every individual system call. Subsequently, various classification algorithms such as the k-nearest neighbour (kNN) [65], [66], k-means clustering (kMC) [67] and support vector machine (SVM) [68] can be applied to differentiate an anomalous frequency vector from the known normal ones using a metric of similarity (or distance). Canzanese *et al.* [64] propose a more complex scheme that combines multiple detectors together for detecting a malicious process. This scheme constructs both ordered and unordered 2-grams from a sequence and adopt the inverse document frequency (TF-IDF) statistics to fabricate the frequency vector. Then, four algorithms signature-based, multinomial log-likelihood ratio test (LLRT), SVM and logistic regression (LR) are implemented in making a decision.

System call-based analytical techniques have also shown great potential in dealing with insider threats. For example, an analysis of system calls relating to file systems and processes is able to address data exfiltration [30]. In particular, the cited scheme is modelling the daily behaviour of accessing files and directory for users and processes respectively. If there is an access to a certain file system location more frequent than an expected range indicated by the model or to an unauthorised location, an anomaly is reported. In the meantime, it reconstructs each process's execution during an extended period as a process tree, according to which once a process forks a child or executes a program that is not on the authorised list or its process tree is largely inconsistent with the baseline, an anomaly is reported too. Liu *et al.* [31] assesses the ways of feature representation in detail, which include n-grams, histograms and parameters. A data set is synthesised with samples affected by typical insider threats such as privilege escalation, malware installation, data exfiltration and violation against data availability. Representations of n-grams and histograms in this case are actually corresponding to the above-mentioned n-gram and frequency models which substantially investigate a system call's sequential information, while representation of parameter is attempting to retrieve information about a system call's parameters and return code. This assessment concludes that representation of parameters is among the most sensitive way. A graph-based scheme is also proposed to cope with insider threats [53], which is motivated by the observation that some system calls are related directly to a user's logon/off and file operations such as *exec, execve, time, login, logout, su, rsh, rexecd, passwd, rexd and ftp*. The graph-based anomaly detection (GBAD) algorithms are implemented to create multiple models (normative structures) for a chunk of system calls and their parameters which, finally, constitute an ensemble to detect subsequent chunks in a streaming manner. During each iteration, a chunk is tested with all the models and a weighted majority voting mechanism is applied to make a decision and, then, the least weighted model is replaced with the new model.

We summarise the taxonomy of system call-based analytics (as shown in Table II) and provide some suggestions. The n-gram based techniques make good use of the temporal correlation appearing in a system call sequence, which evolve from the old-fashioned 'lookahead' to the Markov models characterising state transitions probabilistically and then to the ANNs that deeply exploit the non-linearity. However, apparently, such an evolution comes with consistently increasing computational complexity. The frequency based techniques, on the other hand, are highly scalable but perform well only in a limited number of cases where the majority of the sequence have to be affected by a malicious behaviour. Therefore, they are only considered useful for a quick and coarse analysis. A mix of the two models may achieve a better balance between scalability and effectiveness, which has been discussed in [64]. It is natural to think of that analysis of system calls that reflects user behaviours or critical system activities will yield a capability of addressing insider threats. This has been experimentally validated in [31] which also points out that to reach an acceptable performance it needs additional information such as a system call's parameters (input) and return codes (output). For examples, an access to a sensitive file or directory, a call to a prohibited system call and a creation of an unusual child process are more important in revealing malicious behaviours than a sequence of system call itself. The limitations of system call-based analytics are very evident: 1) system calls only exist in a *nix operating system and their counterparts in Windows operating systems are not available yet (DLLs seem to be a potential solution [70]); and 2) a system call reflects a behaviour at the kernel level, suggesting that a significant amount of effort is required to retrieve information.

B. Command, Keyboard and Mouse

In terms of behavioural biometrics, the *nix shell command has been proven very effective in characterising a user's behaviour [82], [83] and, thus, it finds many applications for detecting masqueraders. The following text illustrates what a sequence of command may look like.

```
pwd, ls, cd, ls, cd, ls, cd, cp, rm
```

Meanwhile, this example has indicated a potentially malicious behaviour that a user copies some data and delete. Since *nix shell command and system call have some properties in common, some system call-based analytical techniques are reused immediately. But, more generally, the schemes are operated within a user-profiling framework, namely: a user profile is learnt from the historical commands in the form of a sequence according to which it predicts whether a sequence of commands is executed by the same user or not. Technically speaking, statistical and machine learning algorithms are still most used although a few others can be seen as well.

The earliest scheme is proposed by Lane and Brodley [71]. Since a fact has been noticed that human users exhibit greater variability than computer programs, in order to gain stronger sensitivity to locally successive matches, this scheme modifies the 'lookhead' scheme [51] with a numeric metric of similarity designed to replace the binary true/false metric. Further, this scheme's data storage capacity is optimised by using the least-recently-used (LRU) pruning strategy and greedy clustering algorithm [72]. The former helps to eliminate those less-used sequences in a user profile, while the latter partitions the sequences into clusters which enables computation of similarity to be counted only on cluster centroids (thus, less memory is required). Coull *et al.* [73] propose a similar scheme that employs the length of the longest common subsequence between two subsequences of command as the metric of similarity. Specifically, a semi-global alignment algorithm is applied to yield the anomalous score for a test sequence, where the scoring mechanism is meant to penalise the number of gaps having to be filled to reach alignment.

Apparently, the above schemes are analogous to the n-gram model except for the use of a fixed size window instead of a sliding window. Therefore, we rename it as sequence model to maintain the generality. The schemes cited below can be regarded as representatives of the frequency model. The first two schemes quantify the abnormality with the Hellinger distance, in which each subsequence of command is transformed into a binary vector according to whether a command occurs historically or not and the one-class SVM classifies the anomalous subsequences [79], [80]. From the user-profiling perspective, a scheme is proposed with the fuzzy concept engaged [81], where a fuzzy user profile is established based on the frequency vector of a subsequence of command. In detail, multiple local user profiles are made to precisely reflect this user's daily behaviour and the periodicity. The fuzzy user profile is simply the combination of all the local ones. When referring to the fuzzy user profile, a 'likelihood' can be assigned to a test command in terms of a membership function and, accordingly, a test sequence's abnormality will be determined by averaging the likelihoods.

Statistical is the other major technical category for analysing commands. For instance, DuMouchel *et al.* [74], [75] attempt to represent a user profile with a one-step transition matrix regarding the executed commands, where detection begins with a null hypothesis stating that a test sequence of command is generated by the same user. These schemes in particular work on every 100 commands (i.e., window size=100), based on Fisher's score [74] and Bayes factor test statistics [75] respectively. As reported by [76], representing a user profile with a high-order Markov chain and its implicative mixture transition distribution (MTD) is able to reflect the transition dynamics more accurately, because of the MTD combining contributions linearly from multiple past steps. When the MTD's parameters are estimated with the Expectation-Maximisationn (EM) algorithm, a likelihood-ratio test is implemented in accepting or rejecting the null hypothesis (same as above). Other than statistical hypothesis testing, Schonlau and Theus designs a simpler test statistics [77] that measures how a test sequence is different from the expected behaviour based on command's uniqueness and popularity. Using the naive Bayes classification, Maxion and Townsend [78] propose a scheme to assign posterior probabilities to every test command in terms of this user's and others' historical data respectively which, then, determines whether a sequence comes from this user or not by examining the cumulative posterior probabilities.

Although not very popular, keyboard and mouse dynamics is also a valid data source for conducting analysis of behavioural biometrics. The best example is the scheme proposed by Ahmed and Traore in 2005 [54], which creates a user profile with features extracted from the dynamic behaviour. The dwell time and flight time are key features for characterising keystroke dynamics, which are known to be very unique for every individual. In regard of mouse dynamics, the average speed against travel distance and movement direction are worked out as the features. An ANN is trained with these features, according to which any inconsistent dynamics will indicate the existence of a masquerader.

Table III shows the taxonomy of command, keyboard and mouse-based analytics. Some techniques derived from system call-based analytics are still applicable but, due to greater variability in human behaviour, the proposed schemes tend more to seek a statistical solution. It is apparent that system call-based analytics have a wider range of applications in addressing

TABLE III
TAXONOMY OF COMMAND, KEYBOARD AND MOUSE BASED ANALYTICS

Threat type	Model	Tech category	Algorithm	Data source
Insider threats	sequence	statistical	sequence match [71] [72]	command sequence
			sequence alignment [73]	
			Fisher's score test [74]	
			Bayes factor test [75]	
			LRT [76]	
			test statistics [77]	
		machine learning	naive Bayes classifier [78]	
	frequency	machine learning	SVM [79] [80]	
		fuzzy	fuzzy user profile [81]	
		machine learning	ANN [54]	keyboard and mouse dynamics

cyber security threats as they can look into any computer programs' (or processes') behaviour at the kernel level. For instance, investigating file operation-related processes is able to address data exfiltration and, analysis of some critical system processes that are known to be vulnerable can help to prevent and detect privilege escalation. Variability is also a noticeable difference between system calls and commands. That is, a sequence of command is often collected for a user on a daily logon/off basis whereas a sequence of system call can only be collected during discrete time periods for a process and, hence, the latter is normally less varied. Moreover, a process generally executes along a predefined routine, resulting in every two successive system calls being heavily dependent on each other. The above two differences have potentially explained why system calls can be handled with a sliding window but a fixed size window is preferred to commands: temporal correlation is essential in dealing with a sequence of system call but modelling accuracy is the key to success in analysing a sequence of command. Although command-based analytical techniques are sometimes interchangeable with system-call based analytics, their limitations are unique: 1) they have to face the difficulty in data collection when more and more user operations are accomplished via a graphic user interface (GUI) rather than a command-line interface (CLI); and 2) keyboard and mouse dynamics seem to be promising and are generally available in modern operating systems but data collection is still challenging.

C. Host Logs

Operating system's logging capability allows to record any events either occur in an operating system or other software/programs run, or messages communicated between different users. They have provided rich data sources for auditing and tracing a host's behaviour and, therefore, are quite suitable for detecting intrusions, malware and malicious insiders. Due to the complexity and redundancy in retrieving useful information from these data sources, they are not yet made use as commonly as the above-mentioned data sources. Furthermore, there is not a common model like n-gram (sequence) or frequency available for analysing host logs. From an algorithm perspective, statistical and machine learning algorithms have found most use cases.

A scheme is proposed to detect intrusions [84] with multiple detectors handling features extracted from Windows performance monitor logs on a one-on-one basis. Each detector fits data with a distribution chosen from a predefined set of candidates such as Gaussian, Erlang, Exponential and Uniform, from which the probability is drawn for a test data instance. When this probability is compared with those drawn from the peer hosts, a significantly deviated one will indicate an intrusion. A final decision is made with a weighted-majority vote mechanism. In the meantime, the detector is tuned incrementally with a labelled data set, which lowers a detector's weight if a false alert is given. Finally, for each host, a ready-to-use ensemble is released which has phased out all the trivial features.

Berlin *et al.* [85] attempt to detect malware using standard Windows audit logs. The data collector is configured to capture all the events regarding file/registry's writes, deletes and executes, and process spawned. After all the events are grouped by process IDs, each process is modelled with its n-grams, all of which are then aggregated into one feature vector that represents the entire raw log. When all the raw logs are transformed into feature vectors, a LR classifier is trained to identify a malicious feature vector.

Relying on a Windows-specific sensor (e.g., RUU) that collects critical system events such as process creations, registry key changes and file system operations, a behavioural biometrics-motivated scheme is proposed to prevent masqueraders and detect malicious insiders [58]. A total of 18 features are extracted from collected events for characterising a user's behaviour such as the numbers of unique processes, registry key queries and files accessed. These features' discriminative capability are assessed via the scalar Fisher's method and Fisher Linear Discriminant (FLD) which in essence are examining each feature's within-class and between-class variance. The Gaussian mixture model (GMM), SVM and kernel density estimation (KDE) are all experimented with a few of the most significant features which, finally, concludes that the GMM outperforms the others.

The taxonomy of host log-based analytics is summarised in Table IV. Since host logs are not easy to collect and in general redundant and very massive, this category of techniques are not thriving yet. However, they are uniquely able to look into data at a higher application level which have provided a concise

TABLE IV
TAXONOMY OF HOST LOG BASED ANALYTICS

Threat type	Model	Tech category	Algorithm	Data source
Intrusion		statistical	ensemble [84]	Win performance monitor logs
Malware	n-gram	statistical	LR [85]	Win audit logs
Insider threats		statistical	GMM, KDE [58]	Win system logs
		machine learning	SVM [58]	

and efficient way of characterising a host or user's behaviour. At the same time, host log-based analytics have not to be restricted in *nix operating systems due to data collection. Except for syslog that is available in *nix operating systems, Windows event log can offer the same information in Windows operating systems. The information redundancy existed in host logs is still the major issue that prevents host-based analytics to be applied more widely, because the process of data cleansing and information retrieving is extremely time consuming.

IV. NETWORK-BASED ANALYTICS

Network core equipments such as routers, switches, load balancers and firewalls all have the ability to collect the network traffic passing through which, traditionally, are considered the major audit data sources for detecting intrusions [86]–[91]. Besides, functioning servers deployed in a network such as Proxy, DHCP, DNS, Active Directory (AD) and Email can be configured with their built-in logging capability to produce additional audit data sources in the form of logs. Some works have identified such network logs' great potential for addressing insider threats [92]–[94]. Intuitively, network logs are more application-specific, adept at characterising how users or hosts behave in a network in terms of a certain service(application), while network traffic provides a wider range of information at a lower network/transport layer.

By definition, network traffic is the data moving across a network at a given point of time and, in computer networks, are mostly encapsulated in packets [95]. There are three major technical categories of data acquisition in the context of network traffic [96]: 1) NetFlow, such as "Cisco NetFlow" and "sFlow", 2) Simple Network Management Protocol (SNMP), such as "MRTG" and "Cricket", and 3) packet sniffers, such as "snoop" and "tcpdump". In theory, all network behaviours and communication patterns can be reconstructed from network traffic, by parsing different packet header fields such as source and destination IP addresses, protocol, source and destination port, and bytes. Some of the relevant analytical techniques can be utilised to deal with cyber security threats immediately. For example, when modelling the sequence of traffic volumes occurred between two entities (e.g., a host and a domain name) as a time series, regression analysis is a powerful tool to identify an anomalous volume which, in reality, is deviated significantly from its expected value [97]. Secondly, conducting a periodicity analysis of communication patterns between a host and a domain name is able to indicate the existence of a beaconing behaviour [98]. Thirdly, unusual or malicious user/host activities are detectable via connectivity analysis [99]. An exemplified application scenario is that a user/host is reported if it connects to many suspicious IP addresses or does not connect to a destination that is supposed.

In some cases, network logs serve as alternative data sources to network traffic. A specific type of network log often represents only a certain application layer protocol or functionality. For example, AD logs collected from a Windows domain controller record only the events regarding user logon/off, permission check and etc. Thus, network logs are generally more formatted and structured, requiring less effort for information retrieval. Furthermore, most network traffic-based analytical techniques can be reapplied to analyse network logs with a little modification. Overall, network log-based analytics are playing an increasingly important role in addressing cyber security threats.

A. Network Traffic

Network traffic-based analytics are traditionally developed for intrusion detection [87], [89], [111]. However, with a little modification, their applications can be easily extended to prevent and detect insider threats. For example, many works have discussed the solutions for tackling botnets. In theory, once the botnet wedged in a victim network has been cleared up, an attacker will have no further chances to exploit the victim network and, hence, no more insider threats posed. The relevant techniques often rely on a specific parser to retrieve information from network traffic for dealing with a certain type of cyber security threats, most of which are actually motivated and experimented with the benchmarked KDD Cup' 99 data set (Computer Network Intrusion Detection' data set) [112]. This data set spans over 9 weeks, encapsulating a total of 4,900,000 connection vectors, each containing 41 features. Four popular categories of attack are simulated and have affected the data set, namely DoS, User-to-Root, Remote-to-Local and probing [113]. These techniques are too broad to be discussed comprehensively. In the following paragraphs, we will introduce only some representative schemes which are closely related to insider threats. In terms of the cited schemes, apart from the commonly seen machine learning and statistical algorithms, rule-based and information entropy-based algorithms are also employed.

One of the earliest schemes [34] takes advantage of the difference of grouping activity between legitimate and Botnet DNS traffic. In particular, DNS traffic is grouped according to domain names and, if two groups of IP address access a same domain name at different times yield a similarity above a threshold, this domain name is labelled as suspicious and inserted into a blacklist. Statistical analysis of DNS traffic is also an effective means for detecting botnets [100]. In the proposed scheme, DDNS (Dynamic DNS) request rates and recurring DDNS replies are extracted as features, with two algorithms developed based on Chebyshev's inequality and Mahalanobis distance respectively for decision-making. As an immediate result of Chebyshev's inequality, a domain name's Canonical DNS request rate (CDRR) is anomalous if

$$P(|CDRR - \mu| > k\sigma) < \frac{1}{k^2}$$

where P stands for probability, μ is the mean and σ the standard deviation. Meanwhile, each domain name's 24 hours' CDRRs are sorted in descending order as a new feature vector. Comparing this feature vector with those obtained from white-listed domain names in terms of Mahalanobis distance, a remarkable difference will suggest an anomalous domain name. DDNS replies are then engaged with the two algorithms to make the scheme complete. Similarly, Internet Relay Chat (IRC)-based botnets are detectable via analysing IRC traffic [102], [103]. Since IRC is an application layer protocol that runs on top of TCP, it only needs to collect TCP flows, which saves a considerable amount of resources. The J48 decision tree, Naive Bayes and Bayesian Network algorithms are implemented for classification of IRC and Non-IRC traffic, leveraging features extracted from a network traffic flow perspective, such as IP protocol, total packets exchanged, duration, average bytes-per-packet, variance of bytes-per-packet and etc. When IRC traffic are correctly identified, a cluster analysis is conducted with 5 additional features that reflect packet inter-arrival times and packet sizes for each IRC flow which, finally, reports the cluster whose size is larger than a threshold as suspicious. Gu *et al.* [101] propose a scheme that aims at IRC and HTTP-based botnets at the same time. Similarly, irrelevant traffic are filtered out, but IRC and HTTP traffic are actually captured with a port-independent matcher rather than classification. Two algorithms are proposed to deal with each group of clients who connect to the same destination (domain name or IP address/port): the response-crowd-density-check algorithm conducts a statistical test (threshold random walk, TRW) to determine whether a group is a part of botnet while the response-crowd-homogeneity-check algorithm clusters each group and determines whether a group is homogeneous in terms of the size of the largest cluster. The Dice coefficient is employed as the metric of similarity between two messages (responses) X and Y, i.e.,

$$\text{Dice}(X, Y) = \frac{2|\text{n}-\text{grams}(X) \cap \text{n}-\text{grams}(Y)|}{|\text{n}-\text{grams}(X)| + |\text{n}-\text{grams}(Y)|}$$

where n represents the size of a sliding window and for a message X with a length of l, $|\text{n}-\text{grams}(X)| = l - n + 1$. If a group is unusually homogeneous, it will be identified as being affected by a botnet. The DISCLOURE [105] is designed to detect a wider range of botnets by analysing Netflow data within a typical machine learning framework. The features are extracted for each group of clients connecting to the same destination, which take into consideration flow characteristics, client access pattern and temporal correlation. Classification is conducted with the J48 decision tree, SVM and random forest algorithms, whereby an anomalous feature vector indicates the existence of a botnet. Al-Bataineh and White [104] focus on a specific scenario where a botnet (Zeus) is stealing data from victims. Data exfiltration is being performed over HTTP traffic, with GET and POST requests encrypted for evading detection. As a result, features are constructed for each HTTP request/response through computing its entropy and Byte Frequency Distribution (BFD). Finally, the J48 decision tree, Naive Bayes and multi-layer perception (MLP) algorithms are applied to classify a botnet.

Rather than disrupting a botnet, some efforts have been made to uncover the actions taken immediately by a malicious insider. In this regard, there is a conceptual framework proposed to detect exfiltration of sensitive data [106]. This framework engages three individual components to analyse outbound network traffic: 1) application identification, 2) content signature generation and detection, and 3) covert communication detection. The first component classifies network traffic into different applications; the second component generates a signature for each classification of application in terms of the contents; and, the last component specifically looks after the communications in which an insider may have intentionally hidden the data being exfiltrated (e.g., by encryption, adding noise, compression and etc.). Still working on outbound network traffic, Fawcett's thesis introduces a more concrete scheme ExFILD to address data exfiletration from an information entropy perspective [107]. The ExFILD extracts sessions from network traffic in light of protocol, data volume and number of packets. For each protocol, a threshold is worked out experimentally by computing entropy for each packet that belongs to a session. When referring to this threshold, any packet and its corresponding session being labelled as anomalous will be logged for further investigation.

In [108], some hands-on practices are reported for tackling APTs by assessing outbound network traffic. Apart from security policies the traffic should comply with, a couple of rules are presented to identify a delivery of RAT and, in the meantime, statistical characteristics of the contaminated traffic are summarised for identifying a botnet. Moreover, they introduce the tools and platforms into which these hands-on practices are potentially able to incorporate. Trend Micro also reports some useful rules (signatures) for detecting various RATs and ongoing APT campaigns in regard of network traffic-based analytics [109]. By looking into DNS and HTTP traffic, the Sandnet is designed for the purpose of detecting malware [110] which, for each extracted feature, details what the expected behaviour is according to experimentation or its statistical characteristics as well as the indication of compromise (IoC).

Table V illustrates the taxonomy of network traffic-based analytics. Overall, the early schemes tend more to utilise protocol-specific traffic individually such as DNS [34], [100], IRC [101]–[103] and HTTP [101], [104]. The newer schemes have adapted to analyse network traffic as a whole for addressing a wider range of threats, which often depend on either a machine learning framework or an ensemble of multiple sub-detectors [105], [108], [109], [114], [115]. Although demolishing a botnet prevents an attacker from posing further threats, the above-mentioned schemes can only be regarded as compromised solutions for addressing insider threats. Instead, analysing outbound network traffic may be able to offer an

TABLE V
TAXONOMY OF NETWORK TRAFFIC BASED ANALYTICS

Threat Type	Tech category	Algorithm	Data source
Insider threats	rule-based	similarity [34]	DNS traffic
		Chebyshev's inequality [100]	
	statistical	Mahalanobis distance [100]	
		LRT [101]	IRC & HTTP traffic
	machine learning	clustering [102] [103] [101]	
		decision tree, Naive Bayes classifier, MLP [104]	
		decision tree, SVM, random forest [105]	Netflow
	conceptual framework	NA [106]	outbound network traffic
	information entropy	decision tree [107]	
APT & malware	rule-based	signature match [108] [109]	
	statistical	correlation analysis [108]	
		correlation analysis [110]	DNS & HTTP traffic

immediate solution against data exfiltration [106], [107]; however, they are facing a challenge in modelling packet payloads which are generally encrypted, compressed or noise added rather than existing in a plain-text format. Due to the complexity of network traffic, machine learning and statistical algorithms are still most widely used and have been proven very effective. Generally speaking, network traffic is a reliable and versatile data source, although the volume is often massive. As a result, efficiency and scalability have to be carefully considered when designing a network traffic-based scheme.

B. Network Logs

To a certain extent, network logs can be regarded as application-specific information parsed from network traffic, as any application layer service is ultimately fulfilled via the network layer. In other words, theoretically network logs can deal with cyber security threats as same as network traffic. Network logs are often collected from the functioning servers deployed in a network, such as Proxy, Email, LDAP, Web server, DHCP, VPN and etc. Technically speaking, there are also some similarities between network logs and network traffic based analytics. For example, statistical and machine learning (deep learning) algorithms are made heavy use. However, since network logs provide information at a higher layer and often in a more formatted and structured manner, they are more appropriate to be fed into a system that comprehensively oversees an enterprise-level network.

Myers *et al.* [118] propose a conceptual framework that deals with a malicious insider who exploits internal organisational Web servers. The key idea is to incorporate monitor and detection capabilities into an existing log management system which is thus enabled to expose unauthorised access and automated activities resulting from a malicious insider by looking into Web server logs. Since graph-based algorithms are suited to represent an insider's correspondence pattern, they have been attempted in addressing insider threats such as [119]. The cited scheme works on Email and Cell phone logs, specifically analysing the correspondence patterns. A normative pattern (a graph substructure) is learnt from the entire graph that describes an insider's correspondences by minimising the description length (MDL) and an incident of interest is raised when a test graph substructure is inconsistent with the normative pattern. Based on Splunk Enterprise (a machine-generated big data platform), Hanley and Montelibano introduce a number of hands-on rules for detecting data exfiltration [117], where Email and Active Directory (AD, or its equivalent LDAP) logs and partial HR records are made use. Each rule is meant to express a specific type of unusual insider activity regarding Email communications. For examples, an insider's daily transferred bytes are beyond a threshold, and the recipient is not found in the organisational name-space. Franc's scheme attempts to detect malicious network traffic by using proxy logs [116]. In terms of source and destination IP addresses, source and destination ports and protocol, events of proxy logs are grouped as flows. Then, the flows are grouped into bags each of which actually represents a pair of user/source IP address and second-level domain. With a total of 15 features extracted form each flow, the SVM and multiple instance learning (MIL) algorithms are implemented in identifying an anomalous bag, indicating an anomalous communication occurred between a user and a domain.

The following works reveal a new trend that correlates and analyses multiple types of network log simultaneously to offer prevention and detection capability at an enterprise-level. A number of prototype systems have been designed, which either integrate a suite of detectors (each works on a specific type of network log) into an ensemble [92], [93], or engage a machine learning framework to deal with a full set of features extracted from multiple types of network log [94], [121]. For example, the PRODIGAL (PROactive Detection of Insider threats with Graph Analysis and Learning) [92], [93] extracts more than 100 features from a wide range of network logs such as Email, proxy, Lightweight Directory Access Protocol (LDAP) and so forth. Multiple detectors are constructed with various statistical, machine learning and graph-based algorithms such as KDE, GMM, LR, kNN, HMM, STINGER, and seed set expansion (SSE) each of which is working with a specific subset of the features. The capabilities of the PRODIGAL are apparently greater than a conventional detector, which generate alerts not only for anomalous observations but also for any unusual patterns and scenarios. A scoring mechanism is employed for

TABLE VI
TAXONOMY OF NETWORK LOG BASED ANALYTICS

Threat Type	Tech category	Algorithm	Data source
Malicious communication	machine learning	SVM, MIL [116]	proxy logs
Data exfiltration	rule-based	signature match [117]	Email, LDAP logs
Insider threats	conceptual framework	NA [118]	Web server logs
	graph-based	MDL [119]	Email, Cell phone logs
		STINGER, SSE [92] [93], IF [120]	Email, proxy , LDAP logs
	statistical	Markov model, LR, KDE, GMM [92] [93]	
	machine learning	KNN [92] [93]	
		kMC [94]	proxy, LDAP, DHCP, VPN logs
	deep learning	DNN, RNN [121]	proxy, Email logs

attributing different anomalies to a same responsible entity (e.g., user, computer, Email address, URL) and assigning a score to this entity according to the number of anomalies. Finally, a manual investigation can be undertaken to identify incidents of interest in light of the business requirements. The Beehive [94] is a system that functions similar to the PRODIGAL but fulfils within a machine learning framework. This system makes use of proxy, DHCP, VPN and LDAP logs at the same time, from which 15 features are extracted for a total of 35,000 hosts on a per day per host basis. These features are determined from three aspects: host, traffic and policy, which are expected to characterise normal activities in an enterprise network. The PCA is applied to reduce the dimensionality of the feature set and then the k-means clustering algorithm is implemented in identifying anomalies. The experiments conducted with a real world data set have demonstrated that the Beehive is very effective against unusual behaviours caused by adware, malware, policy violations and other suspicious activities (subject to manual investigation). There is also a graph-based system proposed to address insider threats [120]. In particular, each user's interactions with devices are modelled as a weighted undirected large-scale bipartite graph, of which the parameters are obtained from relevant LDAP, proxy, Email logs and some other auxiliary information (e.g., file operation and psychological data). The isolation forest (IF) algorithm is applied to detect a suspicious user by looking into the graphs and their sub-graphs. Tuor *et al.*'s scheme [121] once again demonstrates deep learning's exclusive capability in dealing with a large-scale complex machine learning problem. Regarding a user's behaviour, this scheme extracts 408 continuous and 6 categorical features in total from Email and proxy logs and file operations. A deep neutral network (DNN) is trained to produce a series of hidden state vectors which are subsequently fed into a RNN to perform detection in real time where the decision making is actually relied on the conditional probability of the hidden state vector.

Research of network log-based analytics begins almost one decade later than network traffic-based analytics, but have made substantial progress. The taxonomy is shown in Table VI. Except for the scheme proposed in [116] that focuses on identifying malicious communications, most of the proposed schemes and systems aim at addressing insider threats. The conceptual framework [118] and the graph-based [119] and rule-based [117] schemes are representatives of early research in this regard, all of which work with relatively individual types of network log. The more recent systems tend to leverage various types of network log at the same time, particularly tailored to look after an enterprise-level network [92]–[94], [120], [121]. Depending on an ensemble of detectors or a machine learning framework, information are correlated across different types of network log in these systems. For example, an association between a user's job title (LDAP) and the Web categories (proxy) the user has accessed may be potentially exploited in determining an unusual behaviour. In practice, information involved in a network log can be equally obtained from network traffic with a properly designed parser. However, such a parsing process is often extremely time-consuming and, therefore, network logs are sometimes preferred as they have presented the required information in a clear manner. The major disadvantage of network logs is that in order to maintain the data integrity and consistency they often contain redundant information such as duplicate entries and constant text messages, resulting in a large demand for data storage and some extra computational cost.

V. CONTEXTUAL DATA-BASED ANALYTICS

In addition to conventional host and network data, contextual data are being increasingly explored by researchers due to their remarkable use in reducing false positive rate and time taken to make a final decision. In this survey, contextual data refer to those which provide contextual information regarding a human user such as HR and psychological data. According to the literature, it is generally believed that the intent of a user being malevolent can be well captured through contextual data. Usually, HR data are available from an employee directory or a specific ERP (Enterprise Resource Planning) system [125], which can reveal employment related information such as type of employment, remaining years of contract, remaining days of leave, job title, salary range, participated projects, business travel records, performance review and so forth. From HR data, for example, it is easy to find out that en employee has been suffering a stagnant salary increase/promotion for a while, indicating an increased risk for this employee to take malevolent actions [11]. In general, psychological data are not immediately available. Instead, the data collection often needs a specifically designed process of

TABLE VII
TAXONOMY OF CONTEXTUAL DATA BASED ANALYTICS

Threat Type	Tech category	Algorithm	Data source
Insider threats	rule-based	signature match [122]	HR, network
	statistical	KDE [122]	
	graph-based	bipartite graph [123]	HR, host
	conceptual framework	NA [124]	HR, psychological, host, network
	factor analysis	scoring [10]	psychological, host
	graph-based	SAD [11]	psychological, network
	machine learning	Bayesian [11]	

psychological profiling [126] which measures an employee's sentiment changes by analysing the following data: questionnaire [10], social media posts and activities, or dynamics of social connections [11]. Intuitively, a sentimentally unstable or disturbed employee due to disappointment, irritation and/or stress is more likely to take irrational actions against the organisation.

A. HR Data

Two representative works are introduced in this subsection to exhibit how to take advantage of HR data. In general, HR data that contain employment related information are easily accessible in an organisation and potentially have set restrictions on an employee's behaviours which is a kind of critical information for behavioural analysis.

The system ELICIT (Exploit Latent Information to Counter Insider Threats) is proposed to address insider threats based on a 'need-to-know' principle [122], which takes advantage of network traffic and contextual data both. Briefly, this system seeks to identify insiders who abuse the privileges which are not expected within their roles' responsibilities and accountabilities and thus violate the 'need-to-know' principle. Information-use events are generated from network traffic using a series of protocol decoders, into which contextual data collected from an employee directory such as name, office location, job description, seniority and projects are periodically incorporated. When the events are attributed respectively to the users according to the contextual data, a number of hand-coded rules and the KDE based algorithms are employed as sub-detectors. Finally, a threat score is generated for each user via a Bayesian network that aggregates all the alerts triggered by the sub-detectors. Nance and Marty [123] introduce a graph-based scheme to detect insider threats with the basic idea that maps a user's normal behaviours into a bipartite graph according to his/her workgroup role. A large number of precursors are obtained from individual or aggregated entries of various application and operating system logs to represent each workgroup role's normal and expected behaviours. Once a user conducts any out-of-scope behaviours in terms of his/her workgroup role, an alert will be triggered.

B. Psychological Data

Psychological data are not available as commonly as HR data but they are very useful for intent analysis. From a sentiment perspective reliable psychological data will well reflect an employee's feelings and attitude about the organisation and, hence, can be taken as a kind of complementary information for addressing insider threats.

Kandias's scheme attempts combining conventional host-based analytics with psychological profiling [10], which is expected to reduce false positive rate in detecting insider threats. A system call-based analytics is employed in conjunction with an IDS and a honeypot to profiling a computer's usage, while the psychological profiling is attained by using a specifically designed questionnaire that reveals a user's sophistication, predisposition and stress level. When a user's behaviour is suspicious in terms of the host-based analytics, the scheme will seek confirmation from the relevant psychological profile. This is achieved with a three-factor (motive, opportunity and capability) analysis, where a score (low: 1-2, medium: 3-4, high: 5-6) is assigned to each factor for quantifying possibility of being malevolent. The final decision can be made with a simple scoring mechanism that the sum of assigned scores is greater than 8. The CHAMPION (for Columnar Hierarchical Auto-associative Memory Processing In Ontological Networks) is proposed as a conceptual framework to prevent insider threats proactively [124]. This framework aims to alert a malicious action in advance rather than not responding until clear indication of compromise has been observed, by conducting analysis of a user's intent, capability and opportunity which are sort of similar to the three-factor analysis. Network and host-based analytics are applied to uncover any policy violations and unusual access patterns and, from the HR and psychological data, explicit 'human factors' such as correlations between certain personality characteristics and counterproductive work behaviours (or higher-risk employees) are extracted. Taking into consideration the above information entirely, the CHAMPION is able to identify a malicious insider in a timely manner with high confidence. The last example is the scheme proposed by Brdiczka *et al.* [11]. This scheme adopts structural anomaly detection (SAD) to discover anomalies from social and information networks which, in particular, models a user's connectivities in Email communication, social network, Web browsing and etc. as a graph and captures the dynamics that occurs at each node of the graph with the sequential Bayesian algorithm. Then, psychological profiling is leveraged to eliminate false positives through an intent analysis, with three features extracted: motivation, personality and emotional state. A user's connectivity behaviour and the psychological features will be aggregated together to generate a threat score for the user regarding the degree of being malevolent.

Table VII presents the taxonomy of contextual data-based analytics. In terms of data source, we can simply classify the above cited works as 'conventional (host data and/or network data) + HR data' [122], [123], 'conventional + psychological data' [10], [11] and 'conventional + HR and psychological data' [124]. HR data can provide information for setting an employee's expected behaviour and partially for understanding an employee's satisfaction with the employer, while psychological data tend more to reflect an employee's personal characters and sentimental changes. Therefore, HR and psychological data both have found use in intent analysis, and HR data also work for behavioural analysis. Although contextual data-based analytics have shown a great advantage in reducing a false positive rate and saving the time spent on decision-making, some efforts are still needed to make them more technically applicable. Firstly, HR and psychological data often exist in an unstructured and unformatted form. It is hard to clean and retrieve useful information from such messy data. Currently, this is reached with heavy reliance on domain experts' manual works [10], [123]. Secondly, contextual data-based analytics can not work alone without a conventional analytics, for identifying a malicious insider immediately. For example, it is insufficiently evidenced to suggest an employee will take bad actions against the organisation because of stagnant wage growth or negative emotional swings.

VI. Discussion and Research challenges

This section discuss some other aspects regarding this research of preventing and detecting insider threats and present the research challenges that need to be addressed. Figure 3 provides a panoramic mapping between insider threats and their countermeasures.

A. Discussion

1) Publicly Available Data Set: In this subsection, we briefly introduce some publicly available data sets which can be leveraged for research of addressing insider threats. Due to complexity of data collection, volume and privacy concerns, this kind of data sets are still very rare. So far, the Carnegie Mellon University (CMU) CERT Program's insider threat database [127] is the only one that has been released publicly for insider threat research. Within a span of 18 months, this database collects a wide range of data sources such as Email, Proxy and AD logs, file operations, and logon/off, asset, decoy and psychological data. A total of 4000 employees' data are involved, with more than 700 insider threat cases. There are three scenarios. 1) A malicious insider conducts data exfiltration via removable media or a cloud box. 2) A malicious insider delivers malware via removable media to result in sabotage of the ICT system. 3) A masquerader impersonates someone else and conducts data exfiltration via Email. From the data source's perspective, this database has involved host logs, network logs and contextual data and, hence can be applied to develop and test various analytics.

In addition to the CMU's database, for each of the previously mentioned analytics, we present a couple of benchmark data sets (if available). The University of New Mexico (UNM)'s sequence-based intrusion detection data set is built for the system call-based analytics [128]. The system call traces are collected from a number of common privileged programs such as sendmail, ftp, lpr, named and so forth. Some intrusions and/or error conditions are injected during the course of data collection such as privilege escalation and buffer overflow. The Australian defence force academy (ADFA)'s linux data set (ADFA-LD) [129] is a recently released data set that aims at replacing traditional system call data sets with one that better represents a modern computer system. The system call traces are collected while the host (assumed to be a server) is operated as usual and, meanwhile, some popular cyber attacks such as Hydra-FTP, Hydra-SSH, Adduser, Java-Meterpreter, Meter-preter and Webshell are launched against the server. This data set contains around 5000 normal traces for training and validating, and each attack results in 8-20 problematic traces. Unix shell command-based analytics are often experimented with the Purdue's data set [130], which contains 9 sets of sanitised user data generated from 8 Unix users over the course of up to 2 years. The success criteria of a scheme is to differentiate whether two sets of commands are drawn from the same user or not, which is then applicable to detect a masquerader. The KDD' 99 data set that has been mentioned earlier is currently the primary one for experimenting network traffic-based analytics [112]. This data set is collected in the format of TCP dump over a course of 9 weeks, with four main categories of attack injected namely DoS, User-to-Root, Remote-to-Local and probing. Although this data set is no longer widely recognised (almost obsoleted), it still has potential to provide insights into detecting the early stage insider threats. Alternatively, the KDD'99 data set's renovated version NSL-KDD data set [131] (smaller and less redundancy) may have provided an acceptable transition until a more qualified data set is made publicly available.

2) Preventing Insider Threats: One may be aware of that most of the schemes, systems, and conceptual frameworks surveyed above focus primarily on detection rather than prevention. Indeed, a dedicated discussion about preventing insider threats is scarce in the literature. However, from the limited number of references, we still can draw a rough outline by compiling the scattered pieces of materials together. A highly cited guide regarding intrusion detection and prevention systems [132] defines an IDS as software that automates the intrusion detection process and an intrusion prevention system (IPS) is software that fulfils all functionalities of an IDS but also attempts to stop possible incidents. Following this conceptual definition, in the context of addressing insider threats, a prevention scheme is meant to be a detection scheme with additional capabilities to stop insider threats. From this perspective, a detection scheme that has successfully defeated an early stage threat on the intrusion kill chain can be regarded as an effective prevention from an insider threat. More genuine examples are those proactively dealing with insider threats [11], [124] which identify suspicious incidents with a typical detection technique and takes action to

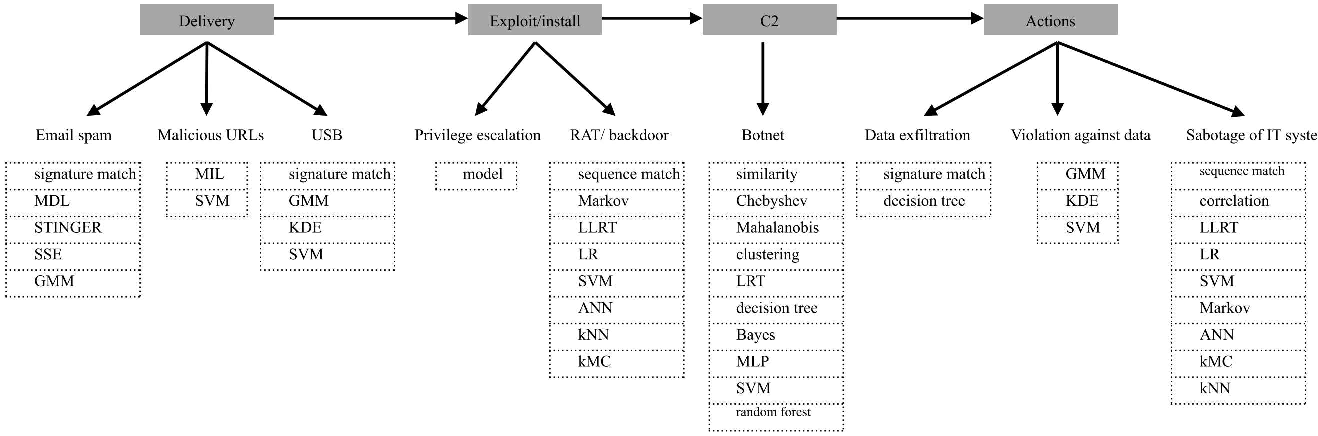


Fig. 3. Insider threats and algorithms.

stop the incidents once they are confirmed from the analysis of psychological profiles.

From a different perspective, prevention can also be thought of as defensive measures that could prevent or facilitate early detection of many of the insider threats [133]. In this regard, the commonly seen measures are comprised of: 1) authentication, 2) access control, and 3) security policy [133]–[135]. Authentication provides the ability to identity entities in a system and between systems and has been traditionally considered as a critical component for preventing insider threats [134]. Its research is being constantly continued with focus on improving the capability against insider threats; for example, a dynamic ID-based remote user authentication scheme [136] is proposed to enable users to choose and change their passwords freely without maintaining any verifier table which has involved design considerations in preventing impersonation from a malicious insider; or, the fourth factor 'where you are' is added on top of the three-factor ('what you have', 'what you know', 'what you are') authentication to address insider threats [137]. Access control is able to restrict access selectively to a place or other resource or, more formally, can be defined as the mechanism providing or limiting access to electronic resources based on some set of credentials [135]. Extensive research has been undertaken in making an access control mechanism insider-aware. For example, a crypto-system engaged with the Key-Policy Attribute-Based Encryption (KPABE) is proposed to implement access control [138]. Due to fine-grained sharing of encrypted data, such a mechanism prevents a malicious insider from stealing and leaking information by accessing data shared among the entire user hierarchy which often occurs in a traditional coarse-grained access control mechanism. Some effort has also been made to remodel new insider-aware access control mechanisms which take into consideration insiderness, trust management and risk assessment [139] and, extend the existing role-based access control mechanism to be insider-aware with the notation of risk and trust integrated [140] where the risk values associated with permissions and roles are calculated using a Coloured Petri-net. Security policy is a definition of what it means to be secure for a system, organisation or other entity, which often exists in the form of a rule (i.e., security policy language) and needs ongoing effort in managing, reviewing and tuning [135]. Practically, there are a number of recommended practices which can be leveraged to enhance the immunity of a system against insider threats [133]; for example, implement strict password and account management policies and deactivate computer access following termination.

Although prevention can be understood in the above two ways and both are critical for addressing insider threats, the former is actually favoured by this survey as we are meant to sort out a clear development roadmap from a data analytics perspective instead of secure system designing.

3) Technical Considerations: This subsection discusses some technical considerations for designing a scheme, from which the skeleton of a scheme can be drawn in a step by step manner. At first, it is not always necessary to focus on an all-in-one scheme, which means that a scheme is designed only in light of the specific use cases and practical requirements and restrictions. For example, when dealing with insiders who are conducting sensitive data leakage, at minimum cost, a scheme only needs to deploy a lightweight statistical detector that works at each local host and analyses unusual file-related operations and network data transfers. In short, three factors matter for a scheme, namely 1) types of insider threat, 2) data sources and 3) analytical techniques. The three factors should be figured out before we begin designing a scheme.

Secondly, we should look at the entity that the analysis being conducted on. Typically, an entity can be user, host, domain name, IP address, program (process), TCP session and any dimension from which a certain behaviour is measurable. Although ultimately we are interested in insider threats, their relevant early stage threats can not necessarily be attributed to a user. For example, when a number of victim hosts have been compromised and controlled by a bot master who refreshes its domain name randomly with the domain generation algorithm (DGA), at this time, a scheme that works on domain name and host may be a better option rather than looking into user's unusual behaviour. However, if we are dealing with malicious insiders who are exfiltrating sensitive data over network, apart from host and TCP session, user should absolutely be inclusive. Obviously, choosing the entity for the analysis is quite open and highly dependent on the use case and application scenario. As a consequence, currently, a more widely applied strategy is to conduct analysis on different entities, aggregate the analytical results into the same entities, and report those top anomalous.

Following the above discussion straightforwardly, we should think about the architecture of a scheme. Based on the cited works, it is easy to summarise that there are two common architectures: 1) engage multiple analytical techniques to work on each data source separately and 2) an individual analytical technique is applied to analyse multiple features extracted from data sources. We can not simply make a conclusion about which architecture outperforms since they both have found a lot of successful uses. It can be observed that it is more flexible to select and tune the analytical technique according to data's specific characteristics when analysing each data source separately, but it fails to take advantage of potential correlations existed between data sources. Working with multiple features at once, on the contrary, is able to provide deeper insights into data although the proposed technique often does not generalise and the computational complexity is significantly increased.

The last paragraph is spent on a brief comparison among behavioural, relationship and intent analyses. Undoubtedly, behavioural analysis is the most widely adopted since behavioural analysis is almost applicable to any behaviour conducted by any entity. For example, we can model each user's Internet browsing behaviour with the following features: number of accessed domain name, number of accessed IP address, amount of sent/received data, number of engaged ports and etc. In contrast, relationship analysis is more suited to discover connectivities among entities such as user-host/user-domain name connections and Email communications, where a graph-based technique is often employed for modelling and inference. Intent analysis can only be conducted on human users, relying on HR or psychological data. It is insufficient to work alone but has shown great potential to eliminate false positives while working with other conventional analytical techniques. Although each of the above three analyses has its own specific application scenarios, recent schemes and systems tend more and more to combine behavioural and intent analyses together.

B. Research Challenges

1) Big Dirty Data Storage and Management: Generally speaking, prevention and detection of insider threats is highly data-driven. That is, we have to deal with a huge amount of data coming from an extremely wide range of computers, servers, network equipments and third party information providers which may exist in an unstructured and unformatted form due to diverse operating systems, data acquisition protocols, configuration issues, hardware faults and software bugs. In this regard, challenges are arisen from how to store and manage such 'big dirty data' properly.

The data we are working with have met the 3Vs definition of big data, namely big volume, high velocity and variety [141]. Thus, from data storage and management perspective, a dedicated big data platform is essential. However, building a big data platform requires massive time and effort and a considerable amount of money, which is not easily affordable. This may be considered as the first challenge. A possible solution is to leverage an open source stack, such as Apache Hadoop [142].

Furthermore, the data are often 'dirty', coming with a large number of unexpected missing and noisy data, duplicate entries, unreadable characters and etc. They have to be pre-processed before useful information can be extracted and passed into the analytics. Such a process relies on extensive scripting and coding skills and a deep understanding about various operating systems, database systems, software and contextual knowledge, resulting in the second challenge "pre-processing of big dirty data". This process can often be accomplished through data cleansing [143] which is supposed to detect and remove corrupt or inaccurate data from a data set, and identify incomplete, incorrect, inaccurate or irrelevant parts of the data and then replace, modify or delete the dirty or corse data [144].

2) Knowledge Extraction and Management: While some of the jobs have been done given that a well-maintained big data platform is in place and the data are pre-processed properly, extraction of useful information for a specific task is still challenging as capturing the tiny footprint left by an attacker is more like "find a needle in a haystack". For example, if we are aiming at detect unusual computer usage, intuitively we may categorise daily computer usage behaviours as logon/off, Email, social media, Web browsing, text editing, professional software operating, video/music playing and so forth. Arranging the categorised behaviours along the time axis yields a model of computer usage according to which the task is achievable. This example raises a requirement that how to extract knowledge about behaviour from raw data sources. In this case, assuming that we have access to any required data sources namely network traffic, AD, proxy, Email logs, and file operation and some other host data, it is still not straightforward to identify those behaviours. Looking at a Web browsing behaviour particularly, we may need to sort out http sessions from the network traffic and, by referring to 1) the mapping between IP address and user from the AD logs and the host data and 2) the http user agent and Web content category information from the proxy logs, determine who the behaviour should be associated with and whether this is certainly a Web browsing behaviour rather than other similar behaviours such as download, software update, or remote desktop access. Technically, such a process of knowledge extraction needs a couple of ways of manipulating data such as evaluating, transforming, correlating and etc., which makes more effective use of data and potentially reduces the amount of data the analytics have to deal with.

However, knowledge extraction has to be repeated if the use case has changed. For example, it may be requested to detect periodic communications between user/host and domain name/IP address, unusual changes of a certain application's state, or unusual authentications. This raises a challenge in managing the extracted knowledge effectively or, in other words, how to improve the reusability of extracted knowledge. The Common Information Model (CIM) [145] is a possible solution to offer stronger reusability for extracted knowledge which is known as "an open standard that defines how managed elements in an IT environment are represented as a common set of objects and relationships between them".

3) Intelligent Decision Making: Choosing an appropriate analytics for making decisions intelligently is always a challenging problem in the context of data analysis. Decision-making is directly related to how to interpret a result from the analytics which, in most of the cases, is simply an binary output: normal or anomalous. As what has been mentioned previously, currently, decision-making is heavily relied on inspections conducted manually by domain experts (e.g., computer emergency response team (CERT) [94]) and being consistently challenged by high false positive rates (FPRs). Due to the excessive amount of data, even a very small number of false positives will result in manual inspection based decision-making hardly feasible. As a consequence, the subsequent research should be focused primarily on removing reliance on manual work (i.e., automated decision-making) and reducing FPRs. Moreover, considering the difficulty in extracting useful knowledge as mentioned in the previous subsection, an automated process of knowledge extraction relying on minimised prior knowledge is also a point making the analytics more intelligent.

Applying a machine/deep learning framework to discover complex dynamics from a wide range of features or leveraging an ensemble of multiple detectors each of which resolves a specific use case is a remarked tendency in the recently proposed schemes and systems [92]–[94], [121]. The former emphasises on automated knowledge extraction and less reliance on domain prior knowledge, whereas sometimes leading to a worse interpretability of the analytics. Incorporating prior knowledge to a certain degree into the procedure of feature extraction may offer a better balance between automation and interpretability. For example, we know that communicating with a domain name that never occurs and grammatically incorrect (e.g., generated by the DGA) may be an indication of botnet. The semantic features of a domain name such as number of pronounceable English words contained and sequential probability in terms of a white/black list should be employed. When these features are involved, naturally, the anomalies resulting from the analytics tend more to reflect the existence of a botnet and, hence, help to reduce the workload of manual inspection as well as the FPRs. The latter comes with better interpretability since a final decision is made by aggregating results from multiple explicit use cases. For example, if we are looking to identify an insider who is exfiltrating sensitive data, we may split this task into a couple of specific use cases: 1) detect unusual file operations based on host data, 2) detect unusual data transfer over network based on network traffic and 3) detect unusual access to sensitive data that does not need to know based on network logs. For each of the use case, a specific detector is developed. Then, a strategy is designed to aggregate the results from the detectors together, popping up the users who are considered suspicious. Such a kind of solutions have taken advantage of domain knowledge, enabling an easier manual inspection to be conducted for making a final decision. But, heavy reliance on prior knowledge also restrict their reusability and extendability and, thus, not sufficiently intelligent. In summary, there is still a long way to make a solution fully independent of prior knowledge and manual inspection which also reaches a sound detection accuracy at an extreme low FPR.

VII. Conclusion

In this survey, we have reviewed the schemes and systems proposed for addressing insider threats. Firstly, we compile the definitions for three major types of insider, namely traitor, masquerader and unintentional perpetrator. Secondly, we conceptually extend the range of insider threat by involving those relevant early stage threats which are all lined up with the APT intrusion kill chain. Then, we focus on the proposed works from a data analytics perspective, where they are presented particularly according to host, network or contextual data based analytics. For each cited work, its capability against insider threats, how it extracts information from data sources and what an analytics/algorithm is applied to make a decision are reviewed. In the meantime, relevant works are compared and contrasted, with a short summary followed to present the pros and cons. Finally we discuss some issues drawn from what we have reviewed and identify a few of research challenges, aiming to motivate and facilitate researchers continuing in contributing to this topic.

References

[1] D. Ding, Q.-L. Han, Y. Xiang, X. Ge, and X.-M. Zhang, "A survey on security control and attack detection for industrial cyber-physical systems," *Neurocomputing*, vol. 275, pp. 1674–1683, Jan. 2018.

[2] *Clearswift Insider Threat Index (CITI)*, Clearswift, Theale, U.K., 2015, accessed: Sep. 6, 2016. [Online]. Available: http://pages.clearswift.com/rs/591-QHZ-135/images/Clearswift_Insider_Threat_Index_2015_US.pdf

[3] D. L. Costa *et al.*, "An insider threat indicator ontology," SEI, Pittsburgh, PA, USA, Rep. CMU/SEI-007, 2016.

[4] M. B. Salem, S. Hershkop, and S. J. Stolfo, "A survey of insider attack detection research," in *Insider Attack and Cyber Security*. Boston, MA, USA: Springer, 2008, pp. 69–90.

[5] CIT Team. (2014). *Unintentional Insider Threats: A Review of Phishing and Malware Incidents by Economic Sector*. Accessed: Nov. 6, 2017. [Online]. Available: https://resources.sei.cmu.edu/asset_files/TechnicalNote/2014_004_001_297777.pdf

[6] M. Collins, *Common Sense Guide to Mitigating Insider Threats*, Carnegie-Mellon Univ., Pittsburgh, PA, USA, 2016.

[7] E. M. Hutchins, M. J. Cloppert, and R. M. Amin, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," presented at the Leading Issues Inf. Warfare Security Res., vol. 1, 2011, p. 80.

[8] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surveys*, vol. 41, no. 3, p. 15, 2009.

[9] S. Wen *et al.*, "A sword with two edges: Propagation studies on both positive and negative information in online social networks," *IEEE Trans. Comput.*, vol. 64, no. 3, pp. 640–653, Mar. 2015.

[10] M. Kandias, A. Mylonas, N. Virvilis, M. Theoharidou, and D. Gritzalis, "An insider threat prediction model," in *Proc. Int. Conf. Trust Privacy Security Digit. Bus.*, Bilbao, Spain, 2010, pp. 26–37.

[11] O. Brdiczka *et al.*, "Proactive insider threat detection through graph learning and psychological context," in *Proc. IEEE Security Privacy Workshops (SPW)*, San Francisco, CA, USA, 2012, pp. 142–149.

[12] D. Kim and M. G. Solomon, *Fundamentals of Information Systems Security*. Burlington, MA, USA: Jones & Bartlett Learn., 2016.

[13] *Penetration Testing: Intelligence Gathering*, InfoSec Inst., Elmwood Park, IL, USA, Jun. 2016, accessed: Nov. 6, 2017. [Online]. Available: http://resources.infosecinstitute.com/penetration-testing-intelligence-gathering/#gref

[14] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Surveying port scans and their detection methodologies," *Comput. J.*, vol. 54, no. 10, pp. 1565–1581, 2011.

[15] S. Jajodia, S. Noel, and B. O'Berry, "Topological analysis of network attack vulnerability," in *Managing Cyber Threats*. Boston, MA, USA: Springer, 2005, pp. 247–266.

[16] J. Bau, E. Bursztein, D. Gupta, and J. Mitchell, "State of the art: Automated black-box Web application vulnerability testing," in *Proc. IEEE Security Privacy (SP)*, Berkeley, CA, USA, 2010, pp. 332–345.

[17] J. Fonseca, M. Vieira, and H. Madeira, "Testing and comparing Web vulnerability scanning tools for SQL injection and XSS attacks," in *Proc. IEEE Depend. Comput.*, Melbourne, VIC, Australia, 2007, pp. 365–372.

[18] C. Hadnagy, *Social Engineering: The Art of Human Hacking*. Hoboken, NJ, USA: Wiley, 2010.

[19] U. Shamir. (2015). *The 7 'Most Common' Rats in Use Today*. Accessed: Nov. 6, 2017. [Online]. Available: https://www.darkreading.com/perimeter/the-7-most-common-rats-in-use-today-/a/d-id/1321965

[20] K. Chiang and L. Lloyd, "A case study of the rustock rootkit and spam bot," in *Proc. HotBots*, vol. 7. Cambridge, MA, USA, 2007, p. 10.

[21] M. Sikorski and A. Honig, *Practical Malware Analysis: The Hands-on Guide to Dissecting Malicious Software*. San Francisco, CA, USA: No Starch Press, 2012.

[22] N. Falliere, L. O. Murchu, and E. Chien, "W32.Stuxnet dossier," Mountain View, CA, USA, Symantec Corporat. Security Response, White Paper, p. 6, 2011.

[23] N. Nissim, A. Cohen, C. Glezer, and Y. Elovici, "Detection of malicious PDF files and directions for enhancements: A state-of-the art survey," *Comput. Security*, vol. 48, pp. 246–266, Feb. 2015.

[24] N. Provos, D. McNamee, P. Mavrommatis, K. Wang, and N. Modadugu, "The ghost in the browser: Analysis of Web-based malware," in *Proc. HotBots*, vol. 7. Cambridge, MA, USA, 2007, p. 4.

[25] L. Xu, Z. Zhan, S. Xu, and K. Ye, "Cross-layer detection of malicious websites," in *Proc. 3rd ACM Conf. Data Appl. Security Privacy*, San Antonio, TX, USA, 2013, pp. 141–152.

[26] S. T. King *et al.*, "Designing and implementing malicious hardware," in *Proc. LEET*, vol. 8. San Francisco, CA, USA, 2008, pp. 1–8.

[27] G. J. Silowash and C. King, "Insider threat control: Understanding data loss prevention (DLP) and detection by correlating events from multiple sources," Softw. Eng. Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, Rep. CMU/SEI-2013-TN-002, 2013.

[28] A. Crenshaw. (2011). *Plug and Prey: Malicious USB Devices*. Accessed: Nov. 6, 2017. [Online]. Available: http://www.irongeek.com/downloads/Malicious%20USB%20Devices.pdf

[29] L. Bilge and T. Dumitras, "Before we knew it: An empirical study of zero-day attacks in the real world," in *Proc. ACM Conf. Comput. Commun. Security*, Raleigh, NC, USA, 2012, pp. 833–844.

[30] N. T. Nguyen, P. L. Reiher, and G. H. Kuenning, "Detecting insider threats by monitoring system call activity," in *Proc. IAW*, West Point, NY, USA, 2003, pp. 45–52.

[31] A. Liu, C. Martin, T. Hetherington, and S. Matzner, "A comparison of system call feature representations for insider threat detection," in *Proc. 6th Annu. IEEE SMC Inf. Assurance Workshop*, West Point, NY, USA, 2005, pp. 340–347.

[32] N. Virvilis and D. Gritzalis, "The big four—What we did wrong in advanced persistent threat detection?" in *Proc. IEEE Availability Rel. Security (ARES)*, Regensburg, Germany, 2013, pp. 248–254.

[33] S. S. C. Silva, R. M. P. Silva, R. C. G. Pinto, and R. M. Salles, "Botnets: A survey," *Comput. Netw.*, vol. 57, no. 2, pp. 378–403, 2013.

[34] H. Choi, H. Lee, H. Lee, and H. Kim, "Botnet detection by monitoring group activities in DNS traffic," in *Proc. IEEE Comput. Inf. Technol.*, Aizuwakamatsu, Japan, 2007, pp. 715–720.

[35] D. Zhao *et al.*, "Botnet detection based on traffic behavior analysis and flow intervals," *Comput. Security*, vol. 39, pp. 2–16, Nov. 2013.

[36] J. B. Grizzard, V. Sharma, C. Nunnery, B. B. Kang, and D. Dagon, "Peer-to-peer botnets: Overview and case study," in *Proc. HotBots*, vol. 7. Cambridge, MA, USA, 2007, p. 1.

[37] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "BotMiner: Clustering analysis of network traffic for protocol-and structure-independent botnet detection," in *Proc. USENIX Security Symp.*, vol. 5. San Jose, CA, USA, 2008, pp. 139–154.

[38] X. Sun, R. Torres, and S. Rao, "DDoS attacks by subverting membership management in P2P systems," in *Proc. IEEE Workshop Secure Netw. Protocols*, Beijing, China, 2007, pp. 1–6.

[39] Y. Xiang, K. Li, and W. Zhou, "Low-rate DDoS attacks detection and traceback by using new information metrics," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 2, pp. 426–437, Jun. 2011.

[40] A. Pathak, F. Qian, Y. C. Hu, Z. M. Mao, and S. Ranjan, "Botnet spam campaigns can be long lasting: Evidence, implications, and analysis," in *Proc. ACM SIGMETRICS Perform. Eval. Rev.*, Seattle, WA, USA, 2009, pp. 13–24.

[41] H. Haddadi, "Fighting online click-fraud using bluff ads," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 2, pp. 21–25, 2010.

[42] D. Plohmann and E. Gerhards-Padilla, "Case study of the miner botnet," in *Proc. Int. Conf. Cyber Conflict*, 2012, pp. 1–16.

[43] Y. Yu and T.-C. Chiueh, "Display-only file server: A solution against information theft due to insider attack," in *Proc. 4th ACM Workshop Digit. Rights Manag.*, Tallinn, Estonia, 2004, pp. 31–39.

[44] X. Huang *et al.*, "Cost-effective authentic and anonymous data sharing with forward security," *IEEE Trans. Comput.*, vol. 64, no. 4, pp. 971–983, Apr. 2015.

[45] J. Baek, Q. H. Vu, J. K. Liu, X. Huang, and Y. Xiang, "A secure cloud computing based framework for big data information management of smart grid," *IEEE Trans. Cloud Comput.*, vol. 3, no. 2, pp. 233–244, Apr./Jun. 2015.

[46] M. Antonakakis, C. Elisan, D. Dagon, G. Ollmann, and E. Wu, *The Command Structure of the Aurora Botnet*, Damballa, Atlanta, GA, USA, 2010.

[47] S. Liu and R. Kuhn, "Data loss prevention," *IT Prof.*, vol. 12, no. 2, pp. 10–13, Mar./Apr. 2010.

[48] A. Gazet, "Comparative analysis of various ransomware virii," *J. Comput. Virol.*, vol. 6, no. 1, pp. 77–90, 2010.

[49] M. Eikel and C. Scheideler, "IRIS: A robust information system against insider DoS attacks," *ACM Trans. Parallel Comput.*, vol. 2, no. 3, p. 18, 2015.

[50] M. Garg. (2006). *Sysenter Based System Call Mechanism in Linux 2.6*. Accessed: Nov. 6, 2017. [Online]. Available: http://articles.manugarg.com/systemcallinlinux2_6.html

[51] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff, "A sense of self for Unix processes," in *Proc. IEEE Security Privacy*, 1996, pp. 120–128.

[52] A. P. Kosoresow and S. A. Hofmeyr, "Intrusion detection via system call traces," *IEEE Softw.*, vol. 14, no. 5, pp. 35–42, Sep./Oct. 1997.

[53] P. Parveen, J. Evans, B. Thuraisingham, K. W. Hamlen, and L. Khan, "Insider threat detection using stream mining and graph mining," in *Proc. IEEE 3rd Int. Conf. Soc. Comput. (SocialCom) Privacy Security Risk Trust (PASSAT)*, Boston, MA, USA, 2011, pp. 1102–1110.

[54] A. A. E. Ahmed and I. Traore, "Anomaly intrusion detection based on biometrics," in *Proc. 6th Annu. IEEE SMC Inf. Assur. Workshop*, West Point, NY, USA, 2005, pp. 452–453.

[55] S. Axelsson, U. Lindqvist, U. Gustafson, and E. Jonsson, "An approach to UNIX security logging," in *Proc. 21st Nat. Inf. Syst. Security Conf.*, 1998, pp. 62–75.

[56] S. Bleha, C. Slivinsky, and B. Hussien, "Computer-access security systems using keystroke dynamics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 12, pp. 1217–1222, Dec. 1990.

[57] A. A. E. Ahmed and I. Traore, "A new biometric technology based on mouse dynamics," *IEEE Trans. Depend. Secure Comput.*, vol. 4, no. 3, pp. 165–179, Jul./Sep. 2007.

[58] Y. Song, M. B. Salem, S. Hershkop, and S. J. Stolfo, "System level user behavior biometrics using fisher features and Gaussian mixture models," in *Proc. IEEE Security Privacy Workshops (SPW)*, 2013, pp. 52–59.

[59] E. Eskin, W. Lee, and S. J. Stolfo, "Modeling system calls for intrusion detection with dynamic window sizes," in *Proc. IEEE DARPA Inf. Survivability Conf. Expo. II*, vol. 1. Anaheim, CA, USA, 2001, pp. 165–175.

[60] J. Hu, X. Yu, D. Qiu, and H.-H. Chen, "A simple and efficient hidden Markov model scheme for host-based anomaly intrusion detection," *IEEE Netw.*, vol. 23, no. 1, pp. 42–47, Jan./Feb. 2009.

[61] A. K. Ghosh, J. Wanken, and F. Charron, "Detecting anomalous and unknown intrusions against programs," in *Proc. IEEE Comput. Security Appl. Conf.*, Phoenix, AZ, USA, 1998, pp. 259–267.

[62] A. K. Ghosh, A. Schwartzbard, and M. Schatz, "Learning program behavior profiles for intrusion detection," in *Proc. Workshop Intrusion Detection Netw. Monitor.*, Santa Clara, CA, USA, 1999, pp. 51–62.

[63] B. Kolosnjaji, A. Zarras, G. Webster, and C. Eckert, "Deep learning for classification of malware system call sequences," in *Proc. Aust. Joint Conf. Artif. Intell.*, 2016, pp. 137–149.

[64] R. Canzanese, S. Mancoridis, and M. Kam, "System call-based detection of malicious processes," in *Proc. IEEE Softw. Qual. Rel. Security (QRS)*, Vancouver, BC, Canada, 2015, pp. 119–124.

[65] Y. Liao and V. R. Vemuri, "Use of k-nearest neighbor classifier for intrusion detection," *Comput. Security*, vol. 21, no. 5, pp. 439–448, 2002.

[66] Y. Liao and V. R. Vemuri, "Using text categorization techniques for intrusion detection," in *Proc. USENIX Security Symp.*, vol. 12, 2002, pp. 51–59.

[67] M. Xie, J. Hu, X. Yu, and E. Chang, "Evaluating host-based anomaly detection systems: Application of the frequency-based algorithms to ADFA-LD," in *Proc. Int. Conf. Netw. Syst. Security*, 2014, pp. 542–549.

[68] M. Xie, J. Hu, and J. Slay, "Evaluating host-based anomaly detection systems: Application of the one-class SVM algorithm to ADFA-LD," in *Proc. IEEE Fuzzy Syst. Knowl. Disc. (FSKD)*, Xiamen, China, 2014, pp. 978–982.

[69] S. A. Hofmeyr, S. Forrest, and A. Somayaji, "Intrusion detection using sequences of system calls," *J. Comput. Security*, vol. 6, no. 3, pp. 151–180, 1998.

[70] W. Haider, G. Creech, Y. Xie, and J. Hu, "Windows based data sets for evaluation of robustness of host based intrusion detection systems (IDS) to zero-day and stealth attacks," *Future Internet*, vol. 8, no. 3, p. 29, 2016.

[71] T. Lane and C. E. Brodley, "An application of machine learning to anomaly detection," in *Proc. 20th Nat. Inf. Syst. Security Conf.*, vol. 377. Baltimore, MD, USA, 1997, pp. 366–380.

[72] T. Lane and C. E. Brodley, "Temporal sequence learning and data reduction for anomaly detection," *ACM Trans. Inf. Syst. Security*, vol. 2, no. 3, pp. 295–331, 1999.

[73] S. Coull, J. Branch, B. Szymanski, and E. Breimer, "Intrusion detection: A bioinformatics approach," in *Proc. IEEE Comput. Security Appl. Conf.*, 2003, pp. 24–33.

[74] W. DuMouchel and M. Schonlau, "A comparison of test statistics for computer intrusion detection based on principal components regression of transition probabilities," in *Proc. Comput. Sci. Stat.*, 1998, pp. 404–413.

[75] W. DuMouchel, "Computer intrusion detection based on Bayes factors for comparing command transition probabilities," AT&T Labs, Nat. Inst. Stat. Sci., Washington, DC, USA, Rep. 91, 1999.

[76] W.-H. Ju and Y. Vardi, "A hybrid high-order Markov chain model for computer intrusion detection," *J. Comput. Graph. Stat.*, vol. 10, no. 2, pp. 277–295, 2001.

[77] M. Schonlau and M. Theus, "Detecting masquerades in intrusion detection based on unpopular commands," *Inf. Process. Lett.*, vol. 76, no. 1, pp. 33–38, 2000.

[78] R. A. Maxion and T. N. Townsend, "Masquerade detection using truncated command lines," in *Proc. Depend. Syst. Netw.*, Washington, DC, USA, 2002, pp. 219–228.

[79] M. B. Salem and S. J. Stolfo, "Detecting masqueraders: A comparison of one-class bag-of-words user behavior modeling techniques," *J. Wireless Mobile Netw. Ubiquitous Comput. Depend. Appl.*, vol. 1, no. 1, pp. 3–13, 2010.

[80] M. B. Salem and S. J. Stolfo, "A comparison of one-class bag-of-words user behavior modeling techniques for masquerade detection," *Security Commun. Netw.*, vol. 5, no. 8, pp. 863–872, 2012.

[81] P. Kudłacik, P. Porwik, and T. Wesołowski, "Fuzzy approach for intrusion detection based on user's commands," *Soft Comput.*, vol. 20, no. 7, pp. 2705–2719, 2016.

[82] B. D. Davison and H. Hirsh, "Predicting sequences of user actions," in *Proc. Notes AAAI/ICML Workshop Predict. Future AI Approaches Time-Series Anal.*, 1998, pp. 5–12.

[83] K. Bhavsar and B. H. Trivedi, "An insider cyber threat prediction mechanism based on behavioral analysis," in *Proc. Int. Conf. ICT Sustain. Develop.*, 2016, pp. 345–353.

[84] J. Shavlik and M. Shavlik, "Selection, combination, and evaluation of effective software sensors for detecting abnormal computer usage," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2004, pp. 276–285.

[85] K. Berlin, D. Slater, and J. Saxe, "Malicious behavior detection using windows audit logs," in *Proc. 8th ACM Workshop Artif. Intell. Security*, 2015, pp. 35–44.

[86] B. Mukherjee, L. T. Heberlein, and K. N. Levitt, "Network intrusion detection," *IEEE Netw.*, vol. 8, no. 3, pp. 26–41, May/Jun. 1994.

[87] M. Roesch *et al.*, "Snort–lightweight intrusion detection for networks," in *Proc. LISA*, vol. 99, no. 1, 1999, pp. 229–238.

[88] S. Northcutt and J. Novak, *Network Intrusion Detection*. Indianapolis, IN, USA: Sams, 2002.

[89] K. Wang and S. J. Stolfo, "Anomalous payload-based network intrusion detection," in *Proc. Int. Workshop Recent Adv. Intrusion Detection*, 2004, pp. 203–222.

[90] J. Zhang, Y. Xiang, W. Zhou, and Y. Wang, "Unsupervised traffic classification using flow statistical properties and IP packet payload," *J. Comput. Syst. Sci.*, vol. 79, no. 5, pp. 573–585, 2013.

[91] J. Zhang *et al.*, "Network traffic classification using correlation information," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 1, pp. 104–117, Jan. 2013.

[92] T. E. Senator *et al.*, "Detecting insider threats in a real corporate database of computer usage activity," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2013, pp. 1393–1401.

[93] W. T. Young, H. G. Goldberg, A. Memory, J. F. Sartain, and T. E. Senator, "Use of domain knowledge to detect insider threats in computer activities," in *Proc. IEEE Security Privacy Workshops (SPW)*, San Francisco, CA, USA, 2013, pp. 60–67.

[94] T.-F. Yen *et al.*, "Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks," in *Proc. 29th Annu. Comput. Security Appl. Conf.*, 2013, pp. 199–208.

[95] K. Park and W. Willinger, *Self-Similar Network Traffic and Performance Evaluation*. New York, NY, USA: Wiley, 2000.

[96] C. So-In, *A Survey of Network Traffic Monitoring and Analysis Tool*, Washington Univ., St. Louis, MO, USA, 2009.

[97] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, 2004, pp. 219–230.

[98] X. Jiang and H. Adeli, "Wavelet packet-autocorrelation function method for traffic flow pattern analysis," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 19, no. 5, pp. 324–337, 2004.

[99] P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," in *Proc. 2nd ACM SIGCOMM Workshop Internet Meas.*, 2002, pp. 71–82.

[100] R. Villamarín-Salomón and J. C. Brustoloni, "Identifying botnets using anomaly detection techniques applied to DNS traffic," in *Proc. IEEE Consum. Commun. Netw. Conf.*, 2008, pp. 476–481.

[101] G. Gu, J. Zhang, and W. Lee, "BotSniffer: Detecting botnet command and control channels in network traffic," in *Proc. 15th Annu. Netw. Distrib. Syst. Security Symp.*, 2008.

[102] W. T. Strayer, R. Walsh, C. Livadas, and D. Lapsley, "Detecting botnets with tight command and control," in *Proc. IEEE Conf. Local Comput. Netw.*, Tampa, FL, USA, 2006, pp. 195–202.

[103] W. T. Strayer, D. Lapsely, R. Walsh, and C. Livadas, "Botnet detection based on network behavior," in *Botnet Detection*. Boston, MA, USA: Springer, 2008, pp. 1–24.

[104] A. Al-Bataineh and G. White, "Analysis and detection of malicious data exfiltration in Web traffic," in *Proc. Malicious Unwanted Softw. (MALWARE)*, 2012, pp. 26–31.

[105] L. Bilge, D. Balzarotti, W. Robertson, E. Kirda, and C. Kruegel, "Disclosure: Detecting botnet command and control servers through large-scale netflow analysis," in *Proc. 28th Annu. Comput. Security Appl. Conf.*, Orlando, FL, USA, 2012, pp. 129–138.

[106] Y. Liu *et al.*, "SIDD: A framework for detecting sensitive data exfiltration by an insider attack," in *Proc. Syst. Sci.*, 2009, pp. 1–10.

[107] T. Fawcett, "ExFILD: A tool for the detection of data exfiltration using entropy and encryption characteristics of network traffic," Ph.D. dissertation, Dept. Elect. Comput. Eng., Univ. Delaware, Newark, DE, USA, 2010.

[108] B. E. Binde, R. McRee, and T. J. O'Connor, "Assessing outbound traffic to uncover advanced persistent threat," Singapore, SANS Inst., White Paper, 2011.

[109] N. Villeneuve and J. Bennett, *Detecting APT Activity With Network Traffic Analysis*, Trend Micro Incorporat., Tokyo, Japan, 2012.

[110] C. Rossow *et al.*, "Sandnet: Network traffic analysis of malicious software," in *Proc. 1st Workshop Build. Anal. Datasets Gathering Exp. Returns Security*, Salzburg, Austria, 2011, pp. 78–88.

[111] J. Zhang, X. Chen, Y. Xiang, W. Zhou, and J. Wu, "Robust network traffic classification," *IEEE/ACM Trans. Netw.*, vol. 23, no. 4, pp. 1257–1270, Aug. 2015.

[112] (1999). SIGKDD Blog. *KDD Cup 1999: Computer Network Intrusion Detection*. Accessed: Jun. 11, 2017. [Online]. Available: http://www.kdd.org/kdd-cup/view/kdd-cup-1999

[113] M. Tavallaee, E. Bagheri, W. Lu, and A.-A. Ghorbani, "A detailed analysis of the KDD cup 99 data set," in *Proc. 2nd IEEE Symp. Comput. Intell. Security Defence Appl.*, Ottawa, ON, Canada, 2009, pp. 1–6.

[114] J. Zhang, C. Chen, Y. Xiang, W. Zhou, and Y. Xiang, "Internet traffic classification by aggregating correlated naive Bayes predictions," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 1, pp. 5–15, Jan. 2013.

[115] Y. Wang *et al.*, "Internet traffic classification using constrained clustering," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 11, pp. 2932–2943, Nov. 2014.

[116] V. Franc, M. Sofka, and K. Bartos, "Learning detector of malicious network traffic from weak labels," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Disc. Databases*, Porto, Portugal, 2015, pp. 85–99.

[117] M. Hanley and J. Montelibano, "Insider threat control: Using centralized logging to detect data exfiltration near insider termination," DTIC, Fort Belvoir, VA, USA, Rep. 024, 2011.

[118] J. Myers, M. R. Grimaila, and R. F. Mills, "Towards insider threat detection using Web server logs," in *Proc. 5th Annu. Workshop Cyber Security Inf. Intell. Res. Cyber Security Inf. Intell. Challenges Strategies*, Oak Ridge, TN, USA, 2009, p. 54.

[119] W. Eberle, J. Graves, and L. Holder, "Insider threat detection using a graph-based approach," *J. Appl. Security Res.*, vol. 6, no. 1, pp. 32–81, 2010.

[120] A. Gamachchi, L. Sun, and S. Boztas, "Graph based framework for malicious insider threat detection," in *Proc. 50th Hawaii Int. Conf. Syst. Sci.*, 2017, pp. 2638–2647.

[121] A. Tuor, S. Kaplan, B. Hutchinson, N. Nichols, and S. Robinson, "Deep learning for unsupervised insider threat detection in structured cybersecurity data streams," in *Proc. AI Cybersecurity Workshop AAAI*, 2017.

[122] M. A. Maloof and G. D. Stephens, "ELICIT: A system for detecting insiders who violate need-to-know," in *Proc. Int. Workshop Recent Adv. Intrusion Detect.*, Gold Coast, QLD, Australia, 2007, pp. 146–166.

[123] K. Nance and R. Marty, "Identifying and visualizing the malicious insider threat using bipartite graphs," in *Proc. Syst. Sci. (HICSS)*, 2011, pp. 1–9.

[124] F. L. Greitzer and R. E. Hohimer, "Modeling human behavior to anticipate insider attacks," *J. Strategic Security*, vol. 4, no. 2, p. 25, 2011.

[125] A. Ragowsky and T. M. Somers, "Enterprise resource planning," *J. Manag. Inf. Syst.*, vol. 19, no. 1, pp. 11–15, 2002.

[126] R. N. Kocsis, H. J. Irwin, A. F. Hayes, and R. Nunn, "Expertise in psychological profiling a comparative assessment," *J. Interpers. Violence*, vol. 15, no. 3, pp. 311–331, 2000.

[127] The CERT Division. *The Cert Insider Threat Database | The Cert Division*. Accessed: Apr. 11, 2017. [Online]. Available: https://www.cert.org/insider-threat/research/database.cfm

[128] *Computer Immune Systems—Data Sets and Software*, Dept. Comput. Sci., Univ. New Mexico, Albuquerque, NM, USA, accessed: Apr. 11, 2017. [Online]. Available: https://www.cs.unm.edu/∼immsec/systemcalls.htm

[129] G. Creech and J. Hu. (2013). *The ADFA Intrusion Detection Datasets*. Accessed: Jun. 11, 2017. [Online]. Available: https://www.unsw.adfa.edu.au/australian-centre-for-cyber-security/cybersecurity/ADFA-IDS-Datasets/

[130] The UC Irvine Machine Learning Repository. *UCI Machine Learning Repository: UNIX User Data Set*. Accessed: Jun. 11, 2017. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/unix+user+data

[131] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani. (2009). *NSL-KDD | Dataset | Research | Canadian Institute for Cybersecurity | UNB*. Accessed: Jun. 11, 2017. [Online]. Available: http://www.unb.ca/cic/datasets/nsl.html

[132] K. Scarfone and P. Mell, *Guide to Intrusion Detection and Prevention Systems (IDPS)*, document 800-94, NIST, Gaithersburg, MD, USA, 2007.

[133] D. Cappelli, A. Moore, R. Trzeciak, and T. J. Shimeall, *Common Sense Guide to Prevention and Detection of Insider Threats 3rd Edition–Version 3.1*, Softw. Eng. Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, 2009. [Online]. Available: http://www.cert.org

[134] R. H. Anderson, "Research and development initiatives focused on preventing, detecting, and responding to insider misuse of critical defense information systems," RAND Corporat., Santa Monica, CA, USA, Rep., 1999.

[135] J. Hunker and C. W. Probst, "Insiders and insider threats–an overview of definitions and mitigation techniques," *J. Wireless Mobile Netw. Ubiquitous Comput. Depend. Appl.*, vol. 2, no. 1, pp. 4–27, 2011.

[136] M. L. Das, A. Saxena, and V. P. Gulati, "A dynamic ID-based remote user authentication scheme," *IEEE Trans. Consum. Electron.*, vol. 50, no. 2, pp. 629–631, May 2004.

[137] S. Choi and D. Zage, "Addressing insider threat using 'where you are' as fourth factor authentication," in *Proc. Security Technol. (ICCST)*, Boston, MA, USA, 2012, pp. 147–153.

[138] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Security*, Alexandria, VA, USA, 2006, pp. 89–98.

[139] J. Crampton and M. Huth, "Towards an access-control framework for countering insider threats," in *Insider Threats in Cyber Security*. Boston, MA, USA: Springer, 2010, pp. 173–195.

[140] N. Baracaldo and J. Joshi, "An adaptive risk management and access control framework to mitigate insider threats," *Comput. Security*, vol. 39, pp. 237–254, Nov. 2013.

[141] A. A. Cardenas, P. K. Manadhata, and S. P. Rajan, "Big data analytics for security," *IEEE Security Privacy*, vol. 11, no. 6, pp. 74–76, Nov./Dec. 2013.

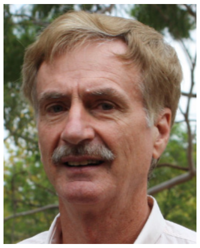
[142] T. Mahmood and U. Afzal, "Security analytics: Big data analytics for cybersecurity: A review of trends, techniques and tools," in *Proc. Inf. Assurance (NCIA)*, 2013, pp. 129–134.

[143] M. A. Hernández and S. J. Stolfo, "Real-world data is dirty: Data cleansing and the merge/purge problem," *Data Min. Knowl. Disc.*, vol. 2, no. 1, pp. 9–37, 1998.

[144] S. Wu, "A review on coarse warranty data and analysis," *Rel. Eng. Syst. Safety*, vol. 114, pp. 1–11, Jun. 2013.

[145] Distributed Management Task Force. *CIM | DMTF*. Accessed: Nov. 11, 2017. [Online]. Available: http://www.dmtf.org/standards/cim

Liu Liu is currently pursuing the Ph.D. degree with the School of Software and Electrical Engineering, Swinburne University of Technology, Australia. Her research interests mainly include malicious insider detection, anomaly detection, deep learning, and cyber security.



Olivier De Vel received the Ph.D. degree in electronic engineering from the Institut National Polytechnique of Grenoble, France. He is currently a Principal Scientist (Cyber) in the Cyber and Electronic Warfare Division, Defence Science and Technology (DST) Group, Department of Defence, Australia. He has worked at several national and international universities, in government research agencies, and in industry research and development laboratories. He joined DST Group in 1999 to set up and provide the scientific research and development leadership in cyber forensics. In 2005, he was appointed as a Research Leader in Cyber Assurance and Operations to lead the DST Group broad spectrum cyber-security program in the C3I Division. His expertise is in the area of artificial intelligence and machine learning for cyber-security and he has published over 100 papers in computer science, digital forensics, and machine learning.

Qing-Long Han (M'09–SM'13) received the B.Sc. degree in mathematics from Shandong Normal University, Jinan, China, in 1983, and the M.Sc. and Ph.D. degrees in control engineering and electrical engineering from the East China University of Science and Technology, Shanghai, China, in 1992 and 1997, respectively. From 1997 to 1998, he was a Post-Doctoral Researcher Fellow with the Laboratoire d'Auomatique et d'Informatique Industrielle (currently, Laboratoire d'Informatique et d'Automatique pour les Systémes), École Supérieure d'Ingénieurs de Poitiers (currently, École Nationale Supérieure d'Ingénieurs de Poitiers), Université de Poitiers, France. From 1999 to 2001, he was a Research Assistant Professor with the Department of Mechanical and Industrial Engineering, Southern Illinois University, Edwardsville, USA. From 2001 to 2014, he was a Laureate Professor, an Associate Dean of research and innovation with the Higher Education Division, and the Founding Director of the Centre for Intelligent and Networked Systems, Central Queensland University, Australia. From 2014 to 2016, he was a Deputy Dean of research with the Griffith Sciences, and a Professor with the Griffith School of Engineering, Griffith University, Australia. In 2016, he joined the Swinburne University of Technology, Australia, where he is currently a Pro Vice-Chancellor of research quality and a Distinguished Professor. In 2010, he was appointed as a Chang Jiang (Yangtze River) Scholar Chair Professor by Ministry of Education, China. His research interests include networked control systems, neural networks, time-delay systems, multiagent systems and complex dynamical systems. He was a recipient of the World's Most Influential Scientific Minds from 2014 to 2016 and the Highly Cited Researcher Award in Engineering according to Thomson Reuters. He is an Associate Editor of a number of international journals including the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE TRANSACTIONS ON CYBERNETICS, and *Information Sciences*.

Jun Zhang received the Ph.D. degree in computer science from the University of Wollongong, Wollongong, Australia, in 2011. He is an Associate Professor with the School of Software and Electrical Engineering, and the Deputy Director of the Swinburne Cybersecurity Laboratory, Swinburne University of Technology, Australia. His research interests include cybersecurity and applied machine learning. He has published over 80 research papers in refereed international journals and conferences, such as the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, the IEEE/ACM TRANSACTIONS ON NETWORKING, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, the ACM Conference on Computer and Communications Security, and the ACM Asia Conference on Computer and Communications Security. He has been internationally recognized as an Active Researcher in cybersecurity, evidenced by his chairing (PC chair, workshop chair, or publicity chair) of eight international conferences from 2013, and presenting of invited keynote addresses in two conferences and an Invited Lecture in IEEE SMC Victorian Chapter.

Yang Xiang (M'07–SM'12) received the Ph.D. degree in computer science from Deakin University, Australia. He is currently a Full Professor and the Dean of Digital Research and Innovation Capability Platform, Swinburne University of Technology, Australia. His research interests include cyber security, which covers network and system security, data analytics, distributed systems, and networking. He is currently leading his team developing active defense systems against large-scale distributed network attacks. He is the Chief Investigator of several projects in network and system security, funded by the Australian Research Council. He has published over 200 research papers in many international journals and conferences. He served as the Associate Editor of the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, *Security and Communication Networks* (Wiley), and the Editor of the *Journal of Network and Computer Applications*. He is the Coordinator, Asia for IEEE Computer Society Technical Committee on Distributed Processing (TCDP).