Journal of Network and Computer Applications 36 (2013) 16–24

Contents lists available at SciVerse ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Review

Intrusion detection system: A comprehensive review

Hung-Jen Liao [a], Chun-Hung Richard Lin [a,*], Ying-Chih Lin [a,b], Kuang-Yuan Tung [a]

[a] Department of Computer Science and Engineering, National Sun Yat-Sen University, No. 70, Lien-hai Rd., 80424 Kaohsiung, Taiwan, ROC
[b] Department of Computer Science and Information Engineering, Cheng Shiu University, No. 840, Cheng-cing Rd., 83347 Kaohsiung, Taiwan, ROC

ARTICLE INFO

Article history:
Received 25 April 2012
Received in revised form
27 August 2012
Accepted 11 September 2012
Available online 23 September 2012

Keywords:
Intrusion detection
Anomaly
Misuse

ABSTRACT

With the increasing amount of network throughput and security threat, the study of intrusion detection systems (IDSs) has received a lot of attention throughout the computer science field. Current IDSs pose challenges on not only capricious intrusion categories, but also huge computational power. Though there is a number of existing literatures to IDS issues, we attempt to give a more elaborate image for a comprehensive review. Through the extensive survey and sophisticated organization, we propose the taxonomy to outline modern IDSs. In addition, tables and figures we summarized in the content contribute to easily grasp the overall picture of IDSs.

Contents

1. Introduction

3W?>Over the past decades, Internet and computer systems have raised numerous security issues due to the explosive use of networks. CERT statistics (CERT) reports that the amount of intrusions has excessively increased year by year. Any malicious intrusion or attack on the network vulnerabilities, computers or information systems may give rise to serious disasters, and violate the computer security policies, i.e., *Confidentiality, Integrity and Availability* (CIA). Up to now, the threats on network and information security are still significant research issues. Though there is a number of existing literatures to survey IDS and its taxonomy (Denning, 1987; Lunt, 1993; Mukherjee et al., 1994; Debar et al., 1999; Axelsson, 2000; Mishra et al., 2004; Krugel and Toth, 2000; Jones and Sielken, 2000; Debar et al., 2000; Mukkamala and Sung, 2003; Estevez-Tapiador et al., 2004; Delgado et al., 2004; Kabiri and Ghorbani, 2005; Anantvalee and Wu, 2007; Patcha and Park, 2007; Tucker et al., 2007; Mandala et al., 2008; Garcia-Teodoro et al., 2009; Amer and Hamilton, 2010; Xie et al., 2011), we try to give a more systematic, architectural and contemporary image for a comprehensive review.

At first, we make a clear distinction about intrusion, intrusion detection, intrusion detection system (IDS) and intrusion prevention system (IPS). NIST (Bace and Mell, 2001) describes the intrusion as an attempt to compromise CIA, or to bypass the security mechanisms of a computer or network, intrusion detection is the process of monitoring the events occurring in a computer system or network, and analyzing them for signs of intrusions. Especially, wireless networks have recently been gaining widespread deployment, and they are much

* Corresponding author. Tel.: +886 7 5252000x4339; fax: +886 7 5254301.
E-mail addresses: hjliao@cse.nsysu.edu.tw (H.-J. Liao), lin@cse.nsysu.edu.tw (C.-H. Richard Lin), yclin@cse.nsysu.edu.tw (Y.-C. Lin), beck@cse.nsysu.edu.tw (K.-Y. Tung).

1084-8045/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jnca.2012.09.004

easier to attack than any wired network. In recent studies (Pelechrinis et al., 2011; Tan et al., 2011), many types of wireless denial of service (WDoS) attacks have been analyzed. Therefore, we categorize IDS into wireless-based and other technology types. The intrusion detection system is the software or hardware system to automate the intrusion detection process (Bace and Mell, 2001; Stavroulakis and Stamp, 2010). Moreover, the intrusion prevention system (IPS) is the system having all IDS capabilities, and could attempt to stop possible incidents (Stavroulakis and Stamp, 2010). In few articles, the terms of intrusion detection and prevention system (IDPS) and IPS are synonyms, where the term IDPS is seldom used in the security community. In this paper, we focus on the survey and classification of IDS related techniques, and give a brief comparison among them.

On the other hand, cloud computing leverages existing technologies, such as virtualization and distributed computing, and has recently emerged as a new paradigm for hosting and delivering services over the Internet. Virtualization is a technology that abstracts away the details of physical hardware and provides the capability of pooling computing resources from clusters of servers, storages and networks for high-level applications. Cloud platforms leverage virtualization technology to achieve the goal of providing computing resources as a utility. Therefore, we also study security issues on *Virtual Machines* (VMs).

The reminder of this paper is organized as follows. We describe IDS methodologies in Section 2, and the classification of IDS approaches in Section 3. Section 4 introduces four classes of IDS technologies. We study IDS issues on VMs in Section 5. Subsequently, two software-oriented solutions, Snort and ClamAV, are studied in Section 6, as they are most widely used open-source tools. Section 7 draws our conclusion, and gives future challenges.

2. Detection methodologies

Intrusion detection methodologies are classified as three major categories: *Signature-based Detection* (SD), *Anomaly-based Detection* (AD) and *Stateful Protocol Analysis* (SPA). Table 1 shows pros and cons of three detection methodologies (Axelsson, 2000; Jones and Sielken, 2000; Debar et al., 2000; Stavroulakis and Stamp, 2010; Lazarevic et al., 2005; Xenakis et al., 2011). Their conceptual descriptions are as follows: signature-based detection (SD)—A signature is a pattern or string that corresponds to a known attack or threat. SD is the process to compare patterns against captured events for recognizing possible intrusions. Because of using the knowledge accumulated by specific attacks and system vulnerabilities, SD is also known as *Knowledge-based Detection* or *Misuse Detection*. Anomaly-based detection (AD)—An anomaly is a deviation to a known behavior, and profiles represent the normal or expected behaviors derived from monitoring regular activities, network connections, hosts or users over a period of time. Profiles can be either static or dynamic, and developed for many attributes, e.g., failed login attempts, processor usage, the count of e-mails sent, etc. Then, AD compares normal profiles with observed events to recognize significant attacks. AD is also called *Behavior-based Detection* in some articles. Some AD's example, e.g., attempted break-in, masquerading, penetration by legitimate user, *Denial-of-Service* (DOS), Trojan horse, etc.

Furthermore, stateful protocol analysis (SPA)—The stateful in SPA indicates that IDS could know and trace the protocol states (e.g., pairing requests with replies). Thought SPA process looks like ADs, they are essentially different. AD adopts preloaded network or host-specific profiles, whereas SPA depends on vendor-developed generic profiles to specific protocols. Generally, the network protocol models in SPA are based originally on protocol standards from international standard organizations, e.g., IETF. SPA is also known as *Specification-based Detection*. *Hybrid*-Most IDSs use multiple methodologies to provide more extensive and accurate detection. For example, SD and AD are complementary methods, because the former concerns certain attacks/threats and the latter focuses on unknown attacks.

3. Detection approaches

Traditionally, people study intrusion detection approaches from two major views, anomaly detection and misuse detection, but there is no considerable difference to their characteristics. Stavroulakis and Stamp (2010) proposed a classification to subdivide these approaches into three subcategories including computation-depended approach, artificial intelligence and biological concepts. However, such a classification is too hard to see the whole properties of detection approaches. Whereas there is a lack of more detailed view for detection approaches, we present a classification of five subclasses with an in-depth perspective on their characteristics: *Statistics-based*, *Pattern-based*, *Rule-based*, *State-based* and *Heuristic-based*. Based on this viewpoint, we carefully marshal current intrusion detection approaches in Table 2.

Time series field in Table 2 indicates whether the mentioned approach considers the time series behavior or not. The type of attacks can be identified by specific approach is presented in the detection of attacks field. Performance field indicates the efficiency at which IDS processes audit events, which has been discussed (Debar et al., 2000; Lazarevic et al., 2005). In addition, type of source contains audit data, user profile, security policies and knowledge extracted from previous attacks. These available data can be used to discriminate intrusion behaviors from suspicious activities. More specialized characteristics for each

Table 1
Pros and cons of intrusion detection methodologies.

Signature-based (knowledge-based)	Anomaly-based (behavior-based)	Stateful protocol analysis (specification-based)
Pros		
• Simplest and effective method to detect known attacks. • Detail contextual analysis.	• Effective to detect new and unforeseen vulnerabilities. • Less dependent on OS. • Facilitate detections of privilege abuse.	• Know and trace the protocol states. • Distinguish unexpected sequences of commands.
Cons		
• Ineffective to detect unknown attacks, evasion attacks, and variants of known attacks. • Little understanding to states and protocols. • Hard to keep signatures/patterns up to date. • Time consuming to maintain the knowledge	• Weak profiles accuracy due to observed events being constantly changed. • Unavailable during rebuilding of behavior profiles. • Difficult to trigger alerts in right time.	• Resource consuming to protocol state tracing and examination. • Unable to inspect attacks looking like benign protocol behaviors. • Might incompatible to dedicated OSs or APs.

Table 2
Classifications and comparisons of various intrusion detection approaches.

	Detection approach	Detection methodology[a]			Time series	Technology type[b]	Detection of attacks[c]	Performance[d]	Type of source	Other characteristics
		AD	SD	SP						
Statistics-based	Statistics (Axelsson, 2000; Debar et al., 2000; Patcha and Park, 2007; Garcia-Teodoro et al., 2009; Xie et al., 2011; Murali and Rao, 2005; Sabahi and Movaghar, 2008; Lazarevic et al., 2005; Fragkiadakis et al., 2012; Mar et al., 2012)	√	√	–	○	H/N	B	M	Audit data, user profiles, usage of disk and memory	Simple but less accuracy
	Distance-based (Patcha and Park, 2007; Murali and Rao, 2005; Sabahi and Movaghar, 2008; Lazarevic et al., 2005)	√	–	–	○	N	U	M	Audit data, network packets	Real-time and active measurement
	Bayesian-based (Kabiri and Ghorbani, 2005; Patcha and Park, 2007; Garcia-Teodoro et al., 2009; Stavroulakis and Stamp, 2010; Lazarevic et al., 2005)	√	√	–	○	N	B	H	Audit data, Prior events, network traffic, user profiles	Optimal statistical (probabilistic) model
	Game Theory (Li et al., 2012; Paramasivan and Pitchai, 2011; Kantzavelou and Katsikas, 2010; Shena et al., 2011)	√	–	–	○	H/N	U	L	System's events or incidents, Log events, byte sent	Self-study, control is poor
Pattern-based	Pattern Matching (Debar et al., 1999; Axelsson, 2000; Krugel and Toth, 2000; Debar et al., 2000; Murali and Rao, 2005; Sabahi and Movaghar, 2008; Lazarevic et al., 2005; Kartit et al., 2012)	–	√	–	×	N	K	H	Audit records, signatures of known attacks	Simple but less flexible
	Perti Net (Debar et al., 1999; Axelsson, 2000; Dexbar et al., 2000; Murali and Rao, 2005; Lazarevic et al., 2005)	–	√	–	○	H	K	M	Audit records, user defined known intrusion signatures	Simple concept and graphic depiction
	Keystroke monitoring (Krugel and Toth, 2000; Murali and Rao, 2005; Lazarevic et al., 2005)	–	√	–	○	H	K	H	Audit records, user profiles, keystroke logs	Using user's typing pattern
	File system checking (Murali and Rao, 2005; Lazarevic et al., 2005)	√	√	–	×	H	B	H	System// configuration/ User files, log files, applications	File integrity checking
Rule-based	Rule-based (Axelsson, 2000; Krugel and Toth, 2000; Jones and Sielken, 2000; Xie et al., 2011; Stavroulakis and Stamp, 2010; Sabahi and Movaghar, 2008; Lazarevic et al., 2005; Farooqi et al., 2012; Modi et al., 2012; Wang et al., 2011)	√	√	–	×	H/N	B	H	Audit records, rule patterns from user profiles and policy	Not easily created and updated
	Data Mining (Kabiri and Ghorbani, 2005; Patcha and Park, 2007; Xie et al., 2011; Murali and Rao, 2005; Lazarevic et al., 2005)	√	√	–	×	N	B	M	Audit data, knowledgebase for association rule discovery	Automatically generated models
	Model/Profile-based (Krugel and Toth, 2000; Murali and Rao, 2005; Sabahi and Movaghar, 2008; Lazarevic et al., 2005; Kartit et al., 2012)	√	–	–	×	H/N	U	M	Audit records, User profiles, Network packets, AP profiles	Varied modeling / profiling methods
	Support vector machine (SVM) (Modi et al., 2012; Kolias et al., 2011; Li et al., 2012; Horng et al., 2011)	√	√	–	○	N	B	H	Limited sample data, binary data	Lower false positive rate, high accuracy
State-based	State-Transition Analysis (Debar et al., 1999; Axelsson, 2000; Krugel and Toth, 2000; Jones and Sielken, 2000; Debar et al., 2000; Stavroulakis and Stamp, 2010; Murali and Rao, 2005; Sabahi and Movaghar, 2008; Lazarevic et al., 2005)	–	√	–	○	H/N	K	H	Audit records, State-transition diagram of known attacks	Flexibility, Detect across user sessions
	User intention Identification (Debar et al., 1999; Debar et al., 2000; Murali and Rao, 2005; Lazarevic et al., 2005)	√	–	–	○	H	U	H	Audit records, user profiles	High-level task pattern
	Markov Process Model (Patcha and Park, 2007; Garcia-Teodoro et al., 2009; Murali and Rao, 2005; Lazarevic et al., 2005; Couture, 2012; Li et al., 2012)	√	–	–	○	H/N	U	M	Audit date, Sequence of system calls or commands.	Probabilistic, Self-training
	Protocol Analysis (Stavroulakis and Stamp, 2010; Murali and Rao, 2005; Sabahi and Movaghar, 2008; Lazarevic et al., 2005)	√	√	√	○	P	T	L	Audit records, Log file, Normal usage (Model) of a protocol	Low false positive rate, Less effective
Heuristic-based	Neural Networks (Axelsson, 2000; Patcha and Park, 2007; Stavroulakis and Stamp, 2010; Murali and Rao, 2005; Lazarevic et al., 2005; Mar et al., 2012; Modi et al., 2012; Wang et al., 2011)	√	√	–	○	N	B	M	Audit data, Sequence of commands, Predict events	Self-learning, Fault tolerant
	Fuzzy Logic (Kabiri and Ghorbani, 2005; Patcha and Park, 2007; Garcia-Teodoro et al., 2009; Stavroulakis and Stamp, 2010; Mar et al., 2012; Modi et al., 2012)	√	–	–	×	H/N	U	H	Audit records, network traffic (TCP/UDP/ICMP)	Configurable, scalable, flexible
	Genetic algorithm (Patcha and Park, 2007; Garcia-Teodoro et al., 2009; Murali and Rao, 2005;	–	√	–	○	N	K	L	Audit data, known attacks	

Table 2 (*continued*)

Detection approach	Detection methodology[a]			Time series	Technology type[b]	Detection of attacks[c]	Performance[d]	Type of source	Other characteristics
	AD	SD	SP						
Lazarevic et al., 2005; Modi et al., 2012; Li et al., 2012; Sen and Clark, 2011)								expressed as binary patterns	Heuristic and evolutionary learning
Immune system (Debar et al., 1999; Debar et al., 2000; Stavroulakis and Stamp, 2010; Murali and Rao, 2005; Lazarevic et al., 2005)	√	√	–	○	H	B	M	Audit data, sequence of system calls	Distributed, high overall security
Swarm Intelligent (SI) (Kolias et al., 2011; Chung and Wahid, 2012; Alomari and Othman, 2012)	√	–	–	○	N	U	H	Network connection data, log file data	Bio-inspired computing intelligence

[a] Detection methodology: anomaly-based detection (AD), signature-base detection (SD), stateful protocol analysis (SP).
[b] Technology type: host-based (H), network-based (N), protocol-based (P).
[c] Detection of attacks: known attacks (K), unknown attacks (U), both known and unknown attacks (B), tripartite of AD, SD and SP (T).
[d] Performance: high (H), moderate (M), low (L).

Table 3
Comparisons of IDS technology types.

Item	Technology			
	HIDS	NIDS	WIDS	NBA
Components[a]	Agent: software (inline) MS: 1∼*n* DS: 1∼*n* (option)	Sensor: *n* (inline/passive) MS: 1∼*n* DS: 1∼*n* (option)	Sensor: *n* (passive) MS: 1∼*n* DS: 1∼*n* (option)	Sensor: *n* (most passive) MS: 1∼*n* (option) DS: optional
Detection scope of sensor/agent	Single host	Network subnet: *n* Host: *n*	WLAN: *n* WLAN client: *n*	Network subnet: *n* Host: *n*
Architecture[b]	MN or SN	MN	MN or SN	MN or SN
Strengths	Only HIDS can analyze end-to-end encrypted communications' activity.	Capable to analyze the broadest scopes of AP protocols	WIDS is more accurate due to its narrow focus. Only WIDS can supervise wireless protocol activity.	Superior detection powers at reconnaissance scanning, reconstruct malware infections and DoS attacks
Technology limitations[c]	• More challenging in detection accuracy due to a lack of context knowledge • Delays in alert generation and centralized reporting • Consume host resources • Conflict with existing security controls	• Cannot monitor wireless protocols • High false positive and false negative rates • Cannot detect attacks within encrypted traffic • No full analysis support under high loads.	• Cannot monitor AL, TL and NL protocol activities. • Cannot avoid evasion techniques. • Sensors are susceptible to physical jamming attacks. • Cannot compensate for insecure wireless protocols	• The major limitation is the delay in detection attacks, caused by transferring flow data to NBA in batches, but not in real time.
Security capabilities				
Information gathering	Network traffic, system calls, file system activity.	Hosts, OSs, APs, network traffic.	WLAN, devices (e.g., APs, clients).	Hosts, OS, services (IP, TCP, UDP, etc).
Logging	Reference (Stavroulakis and Stamp, 2010)	Reference (Stavroulakis and Stamp, 2010)	Reference (Stavroulakis and Stamp, 2010)	Reference (Stavroulakis and Stamp, 2010)
Detection methodology[d]	SD and AD (combined)	SD (major), AD and SPA	AD (major), SD and SPA	AD (major), SPA
Type of suspicious events detected	AL, TL and NL network traffic, event logs (e.g., application activities, file system activities), system logs (e.g., configurations, OS activity)	AL, TL, NL and HW reconnaissance and attacks, unexpected AP services, policy violations	Wireless protocol activity, insecure WLAN and devices, DoS attacks, network scanning, policy violations	AL, TL, NL anomalous traffic flows (DoS attacks, malware) unexpected AP services, network scanning, policy violations

[a] Components: management server (MS), database server (DS).
[b] Network architecture: managed networks (MN), standard networks (SN).
[c] Technology limitations: application (AP), application layer (AL), transport layer (TL), network layer (NL), hardware (HW), operating system (OS).
[d] Detection methodology: signature-based (SD), anomaly-based (AD), stateful protocol analysis (SPA).

mentioned technique are enumerated in other characteristics. In what follows, we give a brief overview for detection approaches.

Statistics-based approaches are mainly by means of predefined threshold, mean and standard deviation, and probabilities to identify intrusions. Pattern-based detection focuses on known attacks through string matching. Moreover, If–Then or If–Then–Else rules are applied in rule-based techniques to construct the model and profile of known intrusions. Specially, state-based methods exploit finite state machine derived from network behaviors to identify attacks. The last one is heuristic-based approach, which is inspired by biological concepts and artificial intelligence. More recent works (Fragkiadakis et al., 2012; Mar et al., 2012; Kartit et al., 2012; Farooqi et al.,2012; Modi et al., 2012; Wang et al., 2011; Couture, 2012; Li et al., 2012) integrate several detection approaches of five subclasses into a sophisticated one to give better efficiency and lower false alarm rate over individual approaches.

4. Technology types

Nowadays, there exist many types of IDS technologies. We categorize the technologies into four classes according to where

they are deployed to inspect suspicious activities, and what event types they can recognize (Mukherjee et al., 1994; Stavroulakis and Stamp, 2010; Sabahi and Movaghar, 2008; Modi et al., 2012). The four classes in Table 3 are as follows: *Host-based IDS* (HIDS), *Network-based IDS* (NIDS), *Wireless-based IDS* (WIDS), *Network Behavior Analysis* (NBA) and *Mixed IDS* (MIDS). An HIDS monitors and collects the characteristics for hosts containing sensitive information, servers running public services, and suspicious activities. An NIDS captures network traffic at specific network segments through sensors, and subsequently, analyzes the activities of applications and protocols to recognize suspicious incidents. WIDS is similar to NIDS, but it captures wireless network traffic, such as ad hoc networks, wireless sensor networks and wireless mesh networks. Besides, an NBA system inspects network traffic to recognize attacks with unexpected traffic flows. Adopting multiple technologies as MIDS can fulfill the goal for a more complete and accurate detection.

Here describes more additional information in Table 3. The Components in IDS include sensor and agent, where the former is typically used for NIDS, WIDS and NBA systems to monitor networks, and HIDS uses the latter to monitor and analyze activities. Both the sensor and agent can deliver data to the *Management Server* (MS) and *Database Server* (DS), where the MS is a centralized device for processing captured incidents, and the DS is just a repository storing event information. Moreover, there are two kinds of network architectures. One is the *Managed Network* (MN), an isolated network deployed for security software management to conceal the IDS information from intruders. MN increases the extra hardware costs and brings about certain inconveniences for administrators. Another is the *Standard Network* (SN), which is a public network without protection. The way to improve SN's security is to build a virtual isolated network by configuring a virtual local area network. On the other hand, most IDS technologies provide four common capabilities for keeping the security, including information gathering, logging, detection and prevention. Information gathering collects information on hosts/networks from observed activities. Logging, the related logging data for detected events, can be used to validate the alerts and investigated incidents. Detection methodologies in most IDSs usually need the sophisticated tuning to receive a higher accuracy. As to the prevention issue, we suggest the reader to refer the survey paper (Stavroulakis and Stamp, 2010) for more excellent expositions.

A common drawback of IDS technologies is that they cannot supply absolutely accurate detection. *False positive* (FP) and *false negative* (FN) are two indicators to assess the degree of accuracy. The former occurs when IDS incorrectly identifies benign activity as being malicious, whereas the latter comes about if IDS fails to identify malicious activity (Stavroulakis and Stamp, 2010; Elshousha and Osmanb, 2011; Shanbhag and Wolf, 2009; Ho et al., 2012). Under the circumstances of failing to have the best of both worlds, many security administrators prefer decreasing FNs to increasing FPs due to the high security consideration. In other words, we may raise more suspicious incidents, and then, distinguish FPs from real suspicious incidents laboriously. More recently, Ho et al. (2012) collect FP and FN cases from real-world traffic, statistically analyze these cases, and propose three findings. First, the great majority of false cases are FNs, because most application behaviors and its content format are self-defined, not conformance to the RFC specifications. Second, most FP alerts are not related to security issues, but to the management policy. Finally, there is an incredibly high percentage of FNs for the aged attacks, including buffer overflow, SQL server attacks and worm slammer attacks.

Furthermore, we summarize and refine many of the previous surveys (Debar et al., 1999, , 2000; Axelsson, 2000; Estevez-Tapiador et al., 2004; Amer and Hamilton, 2010; Bace and Mell, 2001; Sabahi and Movaghar, 2008; Lazarevic et al., 2005; Xenakis et al., 2011) to give a new perspective of taxonomy for IDSs. Figure 1 introduces four aspects to classify IDSs, and the following makes a brief description in sequence. In the branch of *System Deployment*, the *Network Architecture* will be "centralized" that collects and analyzes the information from a single monitored system, "distributed" that collects data from multiple monitored systems so as to detect entire, distributed and cooperative attacks or "hybrid" of both. With state of the art, the distributed configuration should be parallelized, grid-based or cloud-based. The *Networking Type* points out the interconnection of IDS with the system, which is monitored through the fashion of "wired", "wireless" or their "mixed". Especially, wireless IDSs gain explosive requirements, which setup in stand-alone, cooperative or hierarchical environment. The most significant item is *Technology Type*, which has been demonstrated in last section. Second, the facet of *Data Source* which discriminates IDSs based on the system is monitored, and consists of *Collection Component*, i.e., "agent" or "sensor". *Data Collection* via "centralized" or "distributed" gathering. Further, *Data Type* can be (i) audit trails (e.g., system logs, system commands, etc.) on a host, (ii) network packets or connections, (iii) wireless network traffic and (iv) application logs. Third, the *Timeliness* points out that the *Time of Detection* is the "real time/on-line" or "non-real time/off-line" detection for an IDS. In addition, "continuous", "periodic" or "batch" processing for signs of attacks is *Time Granularity*. Further, *Detection Response* to an intrusion has two types: "passive" if an IDS has no countermeasures and only generates alarms; "active" if an IDS takes the corrective or preventive action. Finally, the viewpoint of *Detection Strategy* indicates that the *Detection Discipline* would be "state-based" (secure or insecure) or "transition-based" (from secure to insecure and vice versa), and both of them might be stimulating or non-obtrusive evaluation. Besides, *Processing Strategy* is intuitively "centralized" or "distributed". As to *Detection Methodology*, one of "anomaly-based", "signature-based" and "specification-based" is adopted and illustrated in the previous section.

5. Virtual machines

A virtual machine (VM) (Krutz and Vines, 2010) is a software implementation that emulates a real machine's functionality. Figure 2 is an overview of VM architecture. When the network virtualization isolates virtual networks used by VMs, it also isolates faults and attack impacts in a network. Existing threats, intrusions and attacks to physical and virtual networks are hence minor menaces to VMs (Mosharaf and Boutaba, 2010). However, network virtualization could expose new security vulnerabilities. For example, DoS attacks against the physical network in a virtualized environment will also affect all VMs communicated on the virtual network. An estimation of 60 percent of VMs in production is less secure than their physical counterparts, and 30 percent of deployments with a VM-related security incident (Nikitasha et al., 2011).

Since a VM can be used on-demand, it should be in use at all times; however, the dynamic nature of VMs, known as VM sprawl (Embotics, 2010), makes them difficult to maintain the consistency of security. VMs cloning and migration among physical servers could spread security vulnerabilities and human negligence with an ignorant and rapid way. This will be a disaster against a pool of virtualized servers for production use, because there are generally no physical firewalls separating the VMs in a virtual environment.

Fortunately, most of security concerns have been addressed so that we can prevent most intrusions by applying traditional security defenses to each VM (Zhao et al., 2009). A native method is to assign a dedicated VM to monitor other VMs sharing an identical hypervisor. The monitor can be used in not only IDS, but also integrity checking, honeypot systems and forensic analysis, etc (Payne et al., 2007). Intuitively, this method more or less introduces performance overhead (Xiang et al., 2010). For the intrusion detection inside VMs, *Virtual Memory Introspection* (VMI) (Garfinkel and Rosenblum, 2003)

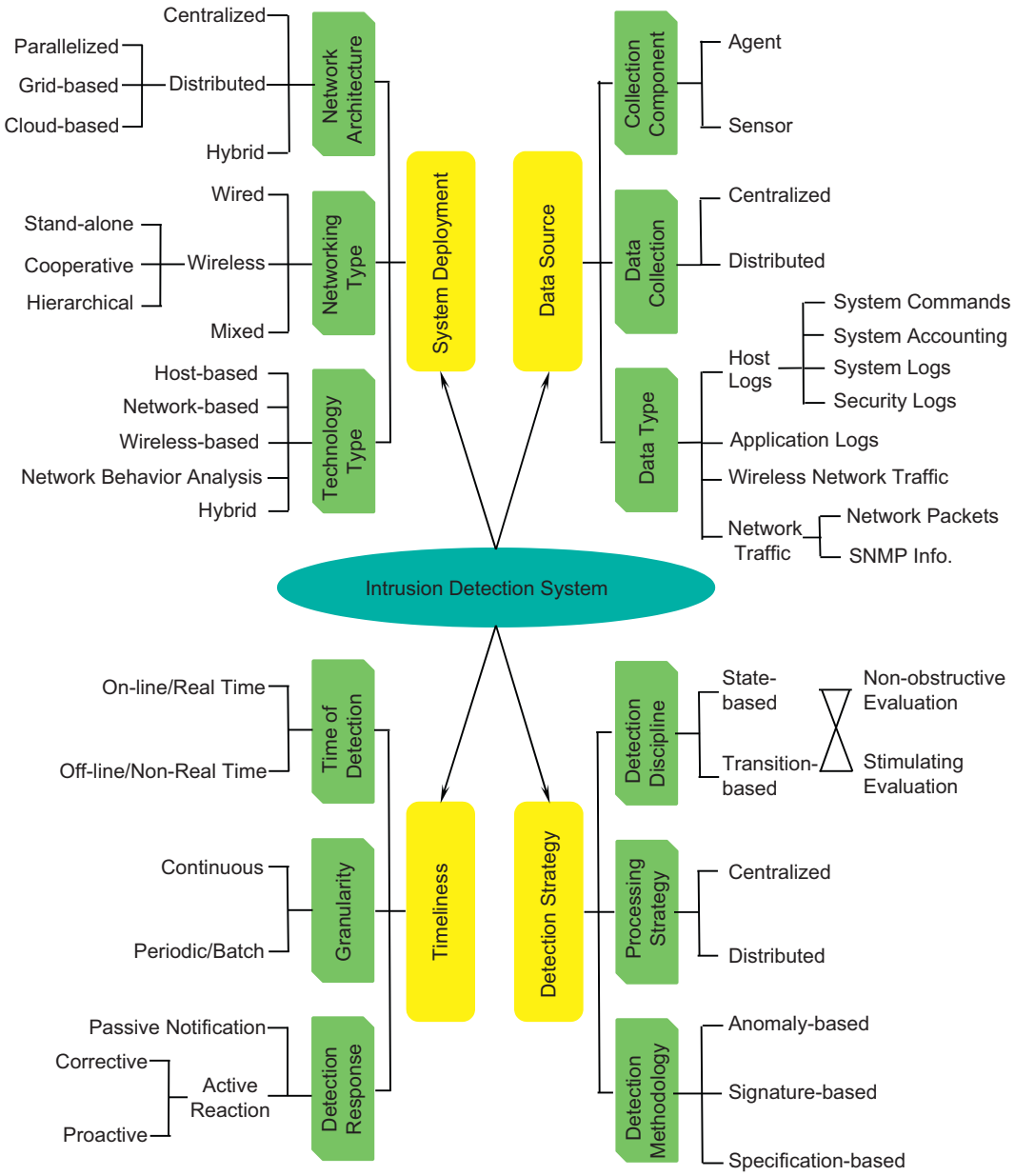

Fig. 1. An overview of IDS taxonomy.

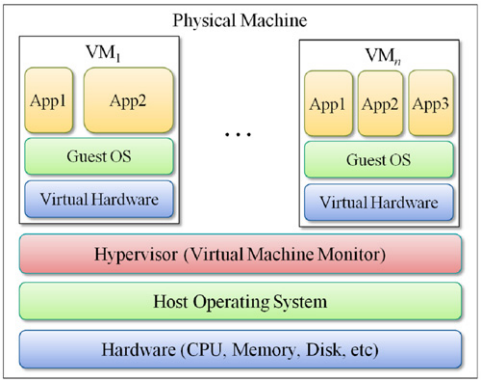

Fig. 2. Virtual machine overview.

is introduced to leverage virtual technology on the hypervisor, while HyperSpector (Kourai and Chiba, 2005) provided a virtual distributed monitoring environment. Besides, deploying IDS on the critical entry of network flow (e.g., load balancer) is also a feasible solution (Reese, 2009).

The hypervisor is a junction of abstracting hardware and allowing host resources sharing between the host and VMs. It is a program running on the host, and hence, susceptible to risk when the volume and complexity of application code increases (Krutz and Vines, 2010). One attack of externally modifying hypervisor is known as VM-based malware/rootkit (VMBR) (Carbone et al., 2008; Le and Wang, 2011), which attempts to execute malicious code instead of system call from hypervisor to the host OS. A *Trusted Platform Module* (TPM) in the host helps to create a trust relationship with the hypervisor (Krutz and Vines, 2010).

6. Snort and ClamAV

High-speed networks and fast-propagating threats pose challenges to current IDSs, which detect break-in attempts by carefully monitoring per packet in the heavy network traffic. Most modern

IDSs possess their own rules whereby they can examine every byte of packets in detail. Here we like to introduce two popular and open source tools implemented by the rule-based approach (Snort; ClamAV). In general, a rule consists of the following elements: A filter specification to what threat of a certain flow the rule works, a string to be the signature of suspicious payloads, a position for the occurrence of that string, and a corresponding action when all the conditions are met.

According to the Amdal's law, string matching would be the first consideration to dramatically improve the performance as it accounts for about 75% CPU load of IDSs (Cabrera et al., 2004). The massive cost comes from the packet check to see whether or not it meets a rule. Though there are many multi-pattern matching algorithms proposed, we cannot afford to examine such a traffic volume against a large set of strings. Moreover, certain signatures are represented in the regular expression to save the storage space, which may need pre-processing techniques to receive significant improvement.

Many works devote attention to the parallel techniques with specialized hardware technologies for improving the packet processing throughput, such as ASIC, Network Processor, FPGA, TCAM, etc (Goyal et al., 2008). These implementations often receive satisfactory performance, e.g., Jiang et al., 2010 claims that their prototype implementation on FPGA sustains 10+ Gbps throughput. However, hardware approaches are usually high-cost, hard to modify, and tied to a specific implementation, which confines their applications.

Due to the drawbacks of the hardware way, some studies look for software-oriented solutions, where Snort and ClamAV are two most widely used open-source tools. The former focuses on the network intrusion detection, whereas the latter is an anti-virus engine. Both have their own signature sets but with a great diversity. Figure 3 shows the length distribution of their signatures. Note that the ClamAV-RE in Fig. 3 represents the signatures with regular expression form in ClamAV, and we do not expand the expression for simplicity. In contrast with the signature set of Snort, ClamAV has more and longer strings. Nowadays, the number of signatures in ClamAV has been over 800,000, and Snort has just a little more than 4000 rules. Even so, Snort's detections could be time-consuming because it examines multi-criteria in a rule.

Snort explores the Aho–Cora (sick algorithm (Aho and Corasick, 1975) for exact-match signature detection; ClamAV uses a variant of the same algorithm to process the signatures with regular expression, and however, the Boyer–Moore algorithm (Boyer and Moore, 1977) to detect the other signatures. There is a considerable rise in the implementation and improvement to both tools. For example, Snort and NTOP, a tool for monitoring network, are combined to form a NIDS in (Peng, 2012). There are

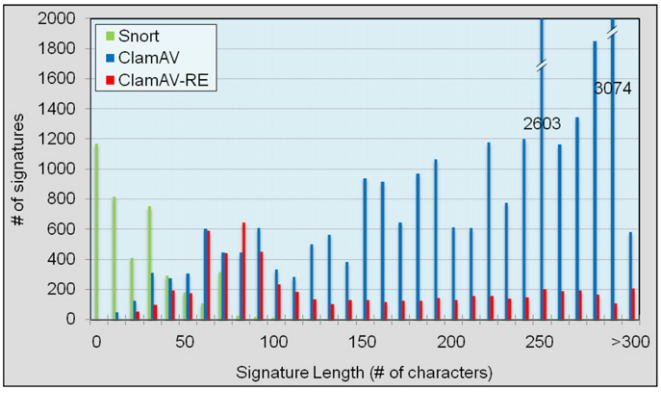

Fig. 3. Distributions of signature lengths in Snort and ClamAV.

works contributed to evaluate Snort performance on Linux and Windows OS (Salah and Kahtani, 2009, 2010; Salah et al., 2011). In addition, Gnort (Vasiliadis et al., 2008) based on Snort is a popular IDS using GPU, and achieves a maximum traffic processing throughput of 2.3 Gbps. Even Gravity (Vasiliadis and Ioannidis, 2010) could up to 20 Gbps, much better than the performance of the CPU-only ClamAV. For the ClamAV pattern set, a memory-saving method is proposed to do the string search for antivirus applications (Wang et al., 2011).

7. Conclusions

7.1. Lessons learned

We have introduced an overview of detection methodologies, approaches and technologies for IDSs. Each technique has its superiority and limitations, so that we should be cautious about selecting the approaches. Take the pattern-based IDS for an instance, although it is simple to implement and very effective to inspect known attacks, the approach could hardly identifies unknown attacks, attacks concealed by evasion techniques and many variants of known attacks. Also, several rule-based approaches to detect unknown attacks have been proposed. However, such techniques may result in the problem of hard creating and updating the knowledge for given attacks. Moreover, heuristic-based approaches have the merit of no prior knowledge of attacks, but do not work well in real-time applications because of the high computational complexity. Therefore, having a comprehensive view of IDSs and application requirements is indispensable before practical usages. In addition, we propose a more elaborate review on IDSs. Tables and figures we summarized contribute to easily grasp the overall picture. Furthermore, we briefly introduce two famous and open-source tools for studying IDSs.

On the other hand, virtualization technology is more and more important as it is extensively used in cloud platforms. The VM is a first virtual component which directly contacts users, and therefore, we also study a number of IDS issues on VMs.

7.2. Future challenges

In this article, we include a comprehensive survey and assessment to current IDSs. However, there remain many open issues and future challenges. For example, *wireless*—due to some particular features (e.g., mobility, no central points, constrained bandwidth of wireless links, and limited resources), the wireless IDSs raise problems about security, communication and management issues. Besides, most wireless IDSs have to be tested under various mobility and topology scenarios for ensuring the protection capacity. *Heuristics*—certain neural, fuzzy and immune-based heuristic IDSs have been proposed, but one should regulate the sensitivity of alerting malicious attacks to decrease false alarm rate. *Parallelism*—high performance computing makes real-time IDSs at low-cost commodity hardware possible; however, there are still many challenges, such as how to divide the jobs of intrusion detections in parallel, the coordination and management of multiple nodes, etc. Moreover, transparent systems, like network filtering facilities, should focus on low-delay processing time, not high-throughput performance. In addition, IDS to VMs with a more slight performance degradation is an urgent topic for services on cloud computing.

References

Aho AV, Corasick MJ. Efficient string matching: an aid to bibliographic search. Communications of the ACM 1975;18:333–40.

Alomari O, Othman ZA. Bees algorithm for feature selection in network anomaly detection. Journal of Applied Sciences Research 2012;8:1748–56.

Amer SH, Hamilton JA. Intrusion detection systems (IDS) taxonomy—a short review. Journal of Software Technology 2010;13.

Anantvalee T, Wu J. A survey on intrusion detection in mobile ad hoc networks. In: Xiao Y, Shen X, Du D-Z, editors. Wireless/mobile network security. Springer-Verlag; 2007. p. 170–96.

Axelsson S, Intrusion detection systems: a survey and taxonomy, Chalmers University of Technology, Sweden, Technical Report 99-15 (2000), pp. 1–27.

R Bace, P Mell, Intrusion detection systems, National Institute of Standards and Technology (NIST), Technical Report 800-31, 2001.

Boyer RS, Moore JS. A fast string searching algorithm. Communications of the ACM 1977;20:762–72.

CERT, 〈http://www.cert.org/stats〉.

Cabrera JBD, Gosar J, Lee W, Mehra RK, On the statistical distribution of processing times in network intrusion detection. In: 43rd IEEE conference on decision and control, Paradise Island, Bahamas, 2004, pp. 75–80.

Carbone M, Lee W, Zamboni D. Taming virtualization. IEEE Security and Privacy 2008;6:65–7.

Chung YY, Wahid N. A hybrid network intrusion detection system using simplified swarm optimization (SSO). Applied Soft Computing 2012;12:3014–22.

ClamAV, 〈http://www.clamav.net〉.

Couture M. Real time intrusion prediction based on optimized alerts with hidden Markov model. Journal of Networks 2012;7:311–21.

Debar H, Dacier M, Wespi A. Towards a taxonomy of intrusion detection systems. Computer Networks 1999;31:805–22.

Debar H, Dacier M, Wespi A. A revised taxonomy for intrusion-detection system. Annals of Telecommunications 2000;55:361–78.

Delgado N, Gates Q, Roach S. A taxonomy and catalog of runtime software-fault monitoring tools. IEEE Transactions on Software Engineering 2004;30:859–72.

Denning DE. An intrusion-detection model. IEEE Transactions on Software Engineering 1987;SE-13:222–32.

Elshousha HT, Osmanb IM. Alert correlation in collaborative intelligent intrusion detection systems—a survey. Applied Soft Computing 2011;11:4349–65.

Embotics, Controlling VM sprawl: best practices for gaining and maintaining control of virtualized infrastructures, White Paper, 2010.

Estevez-Tapiador JM, Garcia-Teodoro P, Diaz-Verdejo JE. Anomaly detection methods in wired networks: a survey and taxonomy. Computer Communications 2004;27:1569–84.

AH Farooqi, FA Khan, J Wang, S Lee, A novel intrusion detection framework for wireless sensor networks, Personal and Ubiquitous Computing. Available Online 2012.

Fragkiadakis AG, Tragos EZ, Tryfonas T, Askoxylakis IG. Design and performance evaluation of a lightweight wireless early warning intrusion detection prototype. EURASIP Journal on Wireless Communications and Networking 2012;73:1–18.

Garcia-Teodoro P, Diaz-Verdejo J, Macia-Fernandez G, Vazquez E. Anomaly-based network intrusion detection: techniques, systems and challenges. Computers & Security 2009;28:18–28.

Garfinkel T, Rosenblum M, A virtual machine introspection based architecture for intrusion detection. In: Network and distributed systems security symposium, San Diego, California, USA, 2003.

Goyal N, Ormont J, Smith R, Sankaralingam K, Estan C, Signature matching in network processing using SIMD/GPU architectures, University of Wisconsin-Madison, Technical Report 1628, 2008.

Ho C-Y, Lai Y-C, Chen I-W, Wang F-Y, Tai W-H. Statistical analysis of false positives and false negatives from real traffic with intrusion detection/prevention systems. IEEE Communications Magazine 2012;50:146–54.

Horng SJ, Su MY, Chen YH, Kao TW, Chen RJ, Lai JL, Perkasa CD. A novel intrusion detection system based on hierarchical clustering and support vector machines. Expert Systems with Applications 2011;38:306–13.

W Jiang, Y-HE Yang, VK Prasanna, Scalable multi-pipeline architecture for high performance multi-pattern string matching. In: 24th IEEE international parallel and distributed processing symposium, Atlanta, Georgia USA, 2010, pp. 1–12.

Jones AK, Sielken RS, Computer system intrusion detection: a survey, University of Virginia, Technical Report (2000).

Kabiri P, Ghorbani AA. Research on intrusion detection and response: a survey. International Journal of Network Security 2005;1:84–102.

Kantzavelou I, Katsikas S. A game-based intrusion detection mechanism to confront internal attackers. Computers & Security 2010;29:859–74.

Kartit A, Saidi A, Bezzazi F, Marraki ME, Radi A. A new approach to intrusion detection system. Journal of theoretical and applied information technology 2012;36:284–9.

Kolias C, Kambourakis G, Maragoudakis M. Swarm intelligence in intrusion detection: a survey. Computers & Security 2011;30:625–42.

Kourai K, Chiba S, HyperSpector: virtual distributed monitoring environments for secure intrusion detection. In: First ACM/USENIX international conference on virtual execution environments, Chicago, IL, USA, 2005, pp. 197–207.

Krugel C, Toth T, A survey on intrusion detection systems, Technical University of Vienna, Austria, Technical Report TUV-1841-00-11 (2000), pp. 22–33.

Krutz RL, Vines RD. Cloud security: a comprehensive guide to secure cloud computing.Indianapolis: Wiley; 2010.

Lazarevic A, Kumar V, Srivastava J. Managing cyber threats: issues, approaches, and challenges.New York: Springer-Verlag; 2005.

Le D, Wang H. An effective memory optimization for virtual machine-based systems. IEEE Transactions on Parallel and Distributed Systems 2011;22:1705–13.

L Li, Zhang G, Nie J, Niu Y, Yao A, The application of genetic algorithm to intrusion detection in MP2P network. In: Third international conference on advances in swarm intelligence, Shenzhen, China, 2012, pp. 390–397.

Li Y, Xia J, Zhang S, Yan J, Ai X, Dai K. An efficient intrusion detection system based on support vector machines and gradually feature removal method. Expert Systems with Applications 2012;39:424–30.

Lunt TF. A survey of intrusion detection techniques. Computers & Security 1993;12:405–18.

Mandala S, Ngadi MA, Abdullah AH. A survey on MANET intrusion detection. International Journal of Computer Science and Security 2008;2:1–11.

Mar J, Hsiao IF, Yeh YC, Kuo CC, Wu SR. Intelligent intrusion detection and robust null defense for wireless networks. International Journal of Innovative Computing Information and Control 2012;8:3341–59.

Mishra A, Nadkarni K, Patcha A, Tech V. Intrusion detection in wireless ad-hoc networks. IEEE Wireless Communications 2004;11:48–60.

C Modi, D Patel, B Borisaniya, H Patel, A Patel, M Rajarajan, A survey of intrusion detection techniques in Cloud. Journal of Network and Computer Applications. Available Online 2012.

Mosharaf NMMK, Boutaba R. A survey of network virtualization. Computer Networks 2010;54:862–76.

Mukherjee B, Heberlein LT, Levitt KN. Network intrusion detection. IEEE Network 1994;8:26–41.

Mukkamala S, Sung AH, A comparative study of techniques for intrusion detection, In: 15th IEEE international conference on tools with artificial intelligence, Sacramento, California, USA, 2003, pp. 570–577.

Murali A, Rao M, A survey on intrusion detection approaches, In: First international conference information and communication technologies, Karachi, Pakistan, 2005, pp. 233–240.

Nikitasha P, Jyotiprakash S, Subasish M, Prasanna PS. A security framework for virtualization based computing environment. International Journal of Engineering Science and Technology 2011;3:6423–9.

Paramasivan B, Pitchai KM. Comprehensive survey on game theory based intrusion detection system for mobile adhoc networks. International Journal of Computing Applications 2011;5:23–9.

Patcha A, Park JM. An overview of anomaly detection techniques: existing solutions and latest technological trends. Computer Networks 2007;51:3448–70.

Payne BD, Carbone M, Lee W, Secure and flexible monitoring of virtual machines, In: 23rd annual computer security applications conference, Miami Beach, FL, 2007, pp. 385–397.

Pelechrinis K, Iliofotou M, Krishnamurthy SV. Denial of service attacks in wireless networks: the case of jammers. IEEE Communications Surveys and Tutorials 2011;13:245–57.

Peng YH, Research of network intrusion detection system based on snort and NTOP, In: Ninth international conference on fuzzy systems and knowledge discovery, Chongqing, China, 2012, pp. 2764–2768.

Reese G. Cloud application architectures: building applications and infrastructure in the cloud. O'Reilly Media; 2009.

Sabahi F, Movaghar A, Intrusion detection: a survey, In: Third international conference on system and network communication, Sliema, Malta, 2008, pp. 23–26.

Salah K, Kahtani A. Improving Snort performance under Linux. IET Communications 2009;3:1883–95.

Salah K, Kahtani A. Performance evaluation comparison of Snort NIDS under Linux and Windows Server. Journal of Network and Computer Applications 2010;33:6–15.

Salah K, Al-Khiaty M-A-R, Ahmed R, Mahdi A. Performance evaluation of Snort under Windows 7 and Windows Server 2008. Journal of Universal Computer Science 2011;17:1605–22.

Sen S, Clark JA. Evolutionary computation techniques for intrusion detection in mobile ad hoc networks. Computer Networks 2011;55:3441–57.

Shanbhag S, Wolf T. Accurate anomaly detection through parallelism. IEEE Network 2009;23:22–8.

Shena S, Li Y, Xua H, Caoa Q. Signaling game based strategy of intrusion detection in wireless sensor networks. Computers & Mathematics with Applications 2011;62:2404–16.

Snort, 〈http://www.snort.org〉.

Stavroulakis P, Stamp M. Handbook of information and communication security.-New York: Springer-Verlag; 2010.

Tan Y, Sengupta S, Subbalakshmi KP. Analysis of coordinated denial-of-service attacks in IEEE 802.22 networks. IEEE Journal on Selected Areas in Communications 2011;29:890–902.

Tucker CJ, Furnell SM, Ghita BV, Brooke PJ. A new taxonomy for comparing intrusion detection systems. Internet Research 2007;17:88–98.

Vasiliadis G, Ioannidis S, GrAVity: a massively parallel antivirus engine, In: Third international conference on recent advances in intrusion detection, Ottawa, Ontario, Canada, 2010, pp. 79–96.

Vasiliadis G, Antonatos S, Polychronakis M, Markatos EP, Ioannidis S, Gnort: high performance network intrusion detection using graphics processors, In: 11th international symposium on recent advances in intrusion detection, Boston, MA, USA, 2008, pp. 116–134.

Wang SS, Yan KQ, Wang SC, Liu CW. An integrated intrusion detection system for cluster-based wireless sensor networks. Expert Systems with Applications 2011;38:15234–43.

Wang X, Wang X, Cao C, Zhu Y. String searching engine for virus scanning. IEEE Transactions on Computers 2011;60:1596–609.

Xenakis C, Panos C, Stavrakakis I. A comparative evaluation of intrusion detection architectures for mobile ad hoc networks. Computers & Security 2011;30:63–80.
Xiang G, Jin H, Zou D, Zhang X, Wen S, Zhao F, VMDriver: a driver-based monitoring mechanism for virtualization, In: 29th IEEE symposium on reliable distributed systems, New Delhi, 2010, pp. 72–81.
Xie M, Han S, Tian B, Parvin S. Anomaly detection in wireless sensor networks: a survey. Journal of Network and Computer Applications 2011;34:1302–25.
Zhao S, Chen K, Zheng W, The application of virtual machines on system security. In: Fourth ChinaGrid annual conference, Yantai, Shandong, 2009, pp. 222–229.