Chapter 1

Intrusion Detection in Wireless Mesh Networks

Thomas M. Chen
Southern Methodist University
Geng-Sheng Kuo
Beijing University of Posts and Telecommunications
Zheng-Ping Li
Beijing University of Posts and Telecommunications
Guo-Mei Zhu
Beijing University of Posts and Telecommunications

Wireless mesh networks are potentially vulnerable to a broad variety of attacks. Hence security is an important consideration for the practical operation of wireless mesh networks. Within security, intrusion detection is the first line of defense in wireless networks as well as wired networks. Unfortunately, wireless mesh networks presents additional challenges due to their decentralized nature, dynamic network topology, and easy access to the radio medium. Due to these unique challenges, intrusion detection techniques can not be borrowed straightforwardly from wired networks. New distributed intrusion detection schemes must be

designed for wireless mesh networks. This chapter describes the basics of intrusion detection and gives a survey of intrusion detection schemes proposed for wireless mesh networks. The schemes share some common concepts but differ in the details which are compared. This chapter describes the difficulties with each scheme and ongoing challenges. Due to the difficult challenges presented by the wireless environment, intrusion detection in wireless mesh networks is still an open research problem.

1.1 Introduction

The main goal of networks is to relay data between its users. Usability in terms of quality of service, availability, and reliability is a typical design objective. The value of a network is perceived by the services it provides to its users. Unfortunately, security is often a secondary consideration and somewhat contradictory to usability. Consequently, many networks are inadequately safeguarded against a variety of attacks. Attackers may use the network to direct attacks at hosts (e.g., to access or control a host), or attackers may aim to damage the network itself.

Attacks are commonplace and readily seen in the Internet today [1]. The average PC user must be aware of good security practices, such as keeping up with operating system patches, running antivirus software, turning on a personal firewall, and avoiding suspicious e-mail attachments. Many of these attacks will eventually cross over to wireless networks as well. For example, many attacks exploit vulnerabilities (weaknesses) in operating systems and applications; these are effective in wired or wireless networks. Also, new types of attacks are evolving constantly.

Typical examples of attacks against hosts include:

- probing for vulnerabilities
- exploiting vulnerabilities to gain unauthorized access
- eavesdropping on communications

- theft or alteration of data
- installation of malicious software (e.g., viruses, worms, Trojan horses, spyware)
- denial of service
- social engineering
- session hijacking.

Some common attacks against the network include:

- denial of service against a router or server
- interception or modification of packets
- interference with routing protocols
- unauthorized tampering of Web, DNS (Domain Name System), or other servers.

Wireless networks are more vulnerable than wired networks because the wireless medium is shared and accessible through the air. In a wired network, an attacker needs to physically access the network to sniff or inject traffic. In a wireless network, an attacker can listen to or transmit packets on a radio link at a distance (and possibly not in visible sight). Thus, the radio medium makes wireless networks both more attractive as targets and harder to defend.

In addition, the mobility afforded by wireless networks is great for users but has certain implications for security. First, mobile devices tend to travel to different, perhaps unfriendly locations. A mobile device is harder to physically secure than a stationary device in a controlled environment. Without adequate physical protection, mobile devices could be physically compromised. Second, mobile users are more difficult to authenticate. A stationary user will always access the network at a known location, so authentication can be based at least in part on location (e.g., a landline phone is identified by its location). A mobile user may access the network at unpredictable locations at different times.

A mobile ad hoc network (MANET) without any fixed infrastructure presents even more challenges for security. With a fixed infrastructure, mobile users could be authenticated with an authentication server that is always accessible regardless of the users location. However, in a MANET with a dynamic network topology, nodes may be disconnected from other nodes for periods of time. A centralized authentication server would not work because it may not be always reachable from a mobile users location.

Without the capability for authentication, impersonation attacks are a major concern in wireless mesh networks. By impersonation, a malicious attacker could participate in the dynamic routing protocol and affect the choice of routes. Wireless mesh networks depend on the cooperation of all nodes to relay packets across the network, so the integrity of the routing protocol is paramount. The effect of an attack on routing could be degradation of network performance, denial of service, or funneling traffic through malicious nodes. Not surprisingly, a great deal of attention has been given to secure routing protocols [2][3][4][5][6][7][8].

A unique type of attack called wormhole has been identified [9]. In physics, a wormhole is theoretically a direct shortcut between two distant points in the space-time continuum. The idea of a wormhole attack is that packets at one location in the network could be tunneled and quickly replayed at another location. A wormhole could be exploited in various ways. For example, it has been hypothesized that routing update packets could go through a wormhole and cause a routing protocol to avoid certain routes [9].

Despite the popular stereotype of a misfit teenage "hacker", there is not a "typical" attacker or a single motive for malicious attacks. An attacker could be almost anyone - a youth looking for fame, a criminal looking for profit, an acquaintance seeking revenge, a competitor attempting industrial espionage, or a hostile foreign military agency. One of the difficulties in network security (both wired and wireless) is the wide range of types of attackers and attack methods.

On the defense side, network security consists of a variety of protective measures usually deployed in a *defense in depth* strategy. Defense in depth refers to multiple lines of defense, such as encryption, firewalls, intrusion detection systems, access controls, antivirus and antispyware programs, combined together to increase the barriers and costs for attackers.

The common belief is that a single perfect defense is not feasible. Instead, an effective deterrent can be constructed from multiple lines of defense, even though each individual element of defense is imperfect. Intrusion detection is one of the most fundamental elements in a defense in depth strategy.

1.2 Intrusion Detection

Intrusion detection can be viewed as a passive defense, similar to a burglar alarm in a building. Unlike firewalls or access controls, intrusion detection systems (IDSs) are not intended to deter or prevent attacks. Instead, their purpose is to alert system administrators about possible attacks, ideally in time to stop the attack or mitigate the damage [10]. Because wireless networks are easier to attack than wired networks, intrusion detection is more critical in wireless networks as a second line of defense.

1.2.1 Goals of Intrusion Detection

Intrusion detection is generally a difficult problem [11]. An IDS attempts to differentiate abnormal activities from normal ones, and identify truly malicious activities (attacks) from the abnormal but non-malicious activities. Unfortunately, normal activities have a wide range, and attacks may appear similar to normal activities. For example, a ping is a common utility to discover if a host is operating and online, but a ping can also be used for attack reconnaissance to learn information about potential targets. Even if unusual activities can be distinguished from normal activities, an unusual activity may not be truly malicious in intent.

The accuracy of intrusion detection is generally measured in terms of false positives (false alarms) and false negatives (attacks not detected). IDSs attempt to minimize both false positives and false negatives. However, this goal is complicated by the likelihood that a skillful attacker will try to evade detection. Thus, detection must be done in adversarial conditions where the attacker may be intelligent and resourceful.

IDSs also attempt to raise alarms while an attack is in progress, so that the attack can be stopped to minimize damage or the attacker can be identified "in the act." This goal is difficult considering that attacks may consist of a sequence of inconspicuous steps; many events (e.g., packets) must be analyzed in real time; and an attack may be new and different from past experiences.

1.2.2 Host-based and Network-based Monitoring

An IDS essentially consists of three functions as shown in Figure 1 [12]. First, an IDS must collect data by monitoring some type of events. IDSs can be classified into two types depending on the monitored events: host-based or network-based IDS. Host-based IDS are installed on hosts and monitor their internal events, usually at the operating system level. These internal events are the type recorded in the hosts audit trails and system logs.

In contrast, network-based IDS monitor packets in the network [13][14][15][16]. This is usually done by setting the network interface on a host to promiscuous mode (so all network traffic is captured, regardless of packet addresses). Alternatively, there are also specialized protocol analyzers designed to capture and decode packets at full link speed.

A popular network-based IDS is the open-source Snort [17]. In its simplest mode, Snort can function as a packet sniffer to view packets traversing a transmission link. In packet logging mode, Snort is able to sniff and dump complete packets into a log for later analysis. Alternatively, Snort configured with a ruleset can function as a real-time IDS. A Snort ruleset is a file of attack signatures that are matched to captured packets. A match to a signature means that an attack is recognized. It is essentially a pattern matching technique. Other popular network-based IDSs are Tcpdump and Ethereal.

The second functional part of an IDS is an analysis engine that processes the collected data. It is programmed with certain intelligence to detect unusual or malicious signs in the data (elaborated below).

The third functional part of an IDS is a response, which is typically an alert to system

administrators. A system administrator is responsible for follow-up investigation of an event after receiving an alert.

1.2.3 Misuse Detection and Anomaly Detection

As mentioned above, the second functional part of an IDS is an analysis engine. Analysis can be done manually by a security expert, but automated analysis is much faster and efficient. The problem with automated analysis is programming the analysis engine with a level of intelligence equivalent to the knowledge and experience of a security expert.

Currently there are two basic approaches to analysis: misuse detection and anomaly detection. Misuse detection is also called signature-based detection because the idea is to represent every attack by a signature (pattern or rule of behavior). Rules can be divided into single part (atomic) signatures or multi-part (composite) signatures. It is essentially a problem of matching the observed traffic to signatures. If a matching signature is found, that attack is detected.

A common implementation of misuse detection is an expert system. An expert system consists of a knowledge base containing descriptions of attack behavior based on past experiences and rules that allow matching of packets against the knowledge base. These rules are often structured as "if-then-else" statements.

An advantage of misuse detection is its accuracy. If a signature matches, that signature identifies the specific attack. Knowledge of the specific type of attack means that an appropriate response can be determined immediately. For its accuracy, misuse detection is widely preferred in commercial systems.

There are two major drawbacks to misuse detection. First, new signatures must be developed whenever a new attack is discovered. Currently, new attacks are evolving constantly. This means that signatures for IDSs must be updated frequently. Second, an attack is recognized only if a matching signature exists. A signature will not exist for new attacks that are significantly different from known attacks. Thus, misuse detection could have a high rate of

false negatives (missed attacks).

Anomaly detection, sometimes called behavior-based detection, is the opposite of misuse detection, as shown in Figure 2. Although they are opposite approaches, they can be used together to realize the advantages of both approaches. Misuse detection tries to characterize attacks, and everything else is assumed to be normal. In contrast, anomaly detection tries to characterize normal behavior, and everything else is assumed to be anomalous (although not necessarily malicious). The underlying premise is that malicious activities will deviate significantly from normal behavior.

The characterization of normal behavior is called a normal profile. A normal profile is usually constructed by statistical analysis of training data. Training data is typically obtained from observations of past normal behavior. Thus, a normal profile is a statistical picture of past normal behavior. Significant deviations from the normal behavior are deemed to be anomalous.

An underlying assumption is that normal behavior will remain the same or at least not change quickly. Since real behavior does change over time, practical anomaly detection systems should adapt the normal profile to track normal behavior changes. This means practical systems should have a capability for automated learning.

A major advantage of anomaly detection is the potential to detect new attacks without prior experience. That is, a signature for a new attack is not required; a new attack will be recognized if it significantly deviates from normal behavior.

There are at least three drawbacks to anomaly detection. First, it has proven to be extremely difficult in practice to accurately characterize normal behavior because normal activities can have large deviations. The choices of statistical metrics for an accurate profile is still an open research problem. Second, anomalous behavior is not necessarily malicious. In fact, a small fraction of anomalous activities may turn out to be an attack. Thus, anomaly detection often shows a high rate of false negatives. These false alarms must be investigated by system administrators, which is time consuming. Third, a detected anomaly does not identify a specific attack, unlike a signature. The lack of specific information means that

system administrators must perform follow-up investigation to determine whether an actual attack is occurring.

1.2.4 IDS Response

As mentioned above, the third functional component of an IDS is the response. Detection of an intrusion must lead to some type of output. Generally, responses can be passive or active. An example of a passive response is to log the intrusion information and raise an alert to system administrators. The IDS does not attempt to impede or stop the intrusion. An IDS response is usually passive because it is widely believed that human judgment (by a trained administrator) is required to formulate the most appropriate course of action. Also, a system administrator often needs to perform further investigation to identify the root cause of an IDS alert.

Active responses attempt to limit the damage of an attack or stop an attack in progress. Damage can be mitigated by protecting the valuable assets or the specific target of the attack. Another active response could be to track the source of the attack, which might be difficult if the attack is being carried out through intermediaries. For example, a distributed denial of service (DDoS) attack is essentially a flooding attack. The flooding traffic usually comes from innocent computers that were surreptitiously compromised by the real attacker. A DDoS attack might be traced to the flooding computers but it is difficult to trace the attack further back.

There is a risk in tying active responses to intrusion detection, an approach called intrusion prevention. In the event of false positives, normal traffic is mistakenly identified as malicious. This would trigger an active response which could cause damage to an innocent user.

1.3 Unique Challenges of Wireless Mesh Networks

Intrusion detection is a common practice in wired networks. Deployment of IDS is well understood and relatively straightforward because the network environment is static. Traffic is relayed by stationary routers. Normally, there are natural points of traffic concentration which are logical candidates for monitoring. For example, private organizations usually connect to the public Internet through a gateway and firewall. All incoming and outgoing traffic go through this point. An IDS just outside of the firewall will be able to see attacks coming from the untrusted Internet. This is informative for understanding the external threats that the firewall is intended to block. Another IDS inside the firewall would monitor the traffic in the private network. If the firewall is effective, no attacks from the outside should be detected. Obviously any detected intrusions means either an insider attack or an external attack penetrated the firewall.

In comparison with wired networks, wireless mesh networks present difficulties for intrusion detection. As a review, wireless mesh networks have sprung from MANETs. MANETs have no fixed infrastructure. All nodes are mobile and the network topology is dynamic. Nodes are simultaneously user devices and routers. The requirements for MANETs have been driven largely by military or specialized civilian applications [18].

Wireless mesh networks relax the requirement of no fixed infrastructure, and can have a mix of fixed and mobile nodes interconnected by wireless links. As in MANETs, mesh nodes can be simultaneously user devices and routers. Nodes might also be fixed wireless routers, e.g., IEEE 802.11 access points, or 802.16 subscriber station [19]. These nodes could constitute a backbone infrastructure [20][21]. A principal characteristic is multihop routing, where packets traverse the network by opportunistic relaying from node to node. Multihop routes through a wireless mesh network are computed by MANET dynamic routing protocols.

1.3.1 Wireless Medium

The wireless medium is one of the major factors effecting intrusion detection. In wired networks, traffic is forced to travel along links, and there are natural points of traffic concentration which are convenient locations for intrusion detection. This is not as valid in a wireless mesh network, particularly if it is entirely ad hoc. However there might be a backbone of fixed wireless routers. In that case, the traffic through access points should be monitored. In practice, this is difficult because access points typically do not have "SPAN ports" that mirror the traffic.

Monitoring traffic by promiscuously eavesdropping on the radio medium is not ideal. Nodes in a wireless mesh network may have relatively short radio ranges (just long enough to reach the next node), so sensors are able to see only limited amounts of traffic. Multiple sensors need to be deployed around the entire network for a comprehensive view of traffic.

Another difficulty presented by the wireless medium is the mobility afforded to nodes. As mentioned earlier, mobile devices might travel to hostile environments. A mobile device without adequate protection could be physically compromised. Therefore, nodes in a wireless mesh network are more vulnerable to compromise and can not be entirely trusted even if their identity is authenticated.

1.3.2 Dynamic Network Topology

Again, the dynamic topology of wireless mesh networks means that there are no natural fixed points of traffic concentration which would be good choices for monitoring.

A possile approach is to run IDS on certain hosts to monitor their local neighborhoods. However, a node can not be expected to monitor the same area for a long time due to its mobility. A node may be unable to obtain a large sample of data for accurate intrusion detection.

1.4 Intrusion Detection for Wireless Mesh Networks

Not surprisingly, most of the research in intrusion detection pertains to MANETs because wireless mesh networks are a relatively recent development. However, virtually all of the intrusion detection schemes for MANETs are relevant to wireless mesh networks.

This section reviews intrusion detection schemes in chronological order to show the evolution of ideas over time up to today. Also, see the survey [22].

1.4.1 WATCHERS

Nodes in a wireless mesh network relay data in a cooperative way that is similar to the way that Internet routers relay IP packets. Therefore, intrusion detection in the Internet environment has direct relevance to intrusion detection in wireless mesh and ad hoc networks. One of the earliest intrusion detection schemes proposed for the Internet environment was WATCHERS (Watching for Anomalies in Transit Conservation: a Heuristic for Ensuring Router Security) [23]. Although WATCHERS was not specifically intended for ad hoc networks, all nodes in ad hoc networks function as routers so the WATCHERS approach is easily applicable. Later intrusion detection schemes for ad hoc networks have followed similar ideas from WATCHERS.

WATCHERS assumes a *wired* mesh network of routers where individual routers may be compromised by an attacker, or malfunctioning due to a fault or misconfiguration. In either case, it is assumed that an intrusion or malfunction will be manifested in the router's misbehavior (selectively dropping or misrouting packets) that can be observed by other routers.

One of the important ideas of WATCHERS is a totally distributed intrusion detection scheme running concurrently and independently in every router. Each router checks incoming packets to detect any routing anomalies. Also, each router keeps track of the amount of data going through neighboring routers. The objective is to detect misbehaving routers in a distributed way.

A link-state routing protocol is assumed. This assumption is necessary so that each router is aware of other routers and the overall network topology. Each router counts any packets that are misrouted by neighboring routers, based on knowledge of their neighbors' routing tables from the link-state routing protocol. Each router also keeps count of the amount of data received and transmitted on all interfaces.

Routers periodically share their respective data by a flooding protocol, and then start a diagnostic phase. Flooding is necessary to overcome any malicious nodes that might try to interfere in the information sharing by blocking packets. In the diagnostic phase, the counts collected from all routers are compared to determine if any routers (1) have misrouted too many packets (2) have not participated correctly in the WATCHERS scheme (3) broadcasted counts that have discrepancies with the counts from their neighbors (4) have appeared to drop more packets than a given threshold. If a router is found to exhibit any of these misbehaviors, it is deemed to be a bad router (but it is impossible to determine if the cause is an intrusion or malfunction, based solely on the router's external behavior). In response to any routers deemed to be misbehaving, routing tables at good routers are changed to avoid forwarding packets through those misbehaving routers.

The counts are compared to thresholds. In an ideal world, the thresholds would be zero, but in practice, the thresholds should be chosen to be more than zero. For example, even good routers may drop a significant number of packets if the router is congested. Therefore, the threshold for number of dropped packets could be high. The choice of proper thresholds can be difficult. If the thresholds are too high, misbehaving routers could be undetectable. On the other hand, if thresholds are too low, the rate of false alarms could be significant.

There are costs involved in the WATCHERS scheme. Each router must use memory to keep counts and a routing table for each neighboring router. Also, all routers are involved in a flooding protocol to share information before each diagnostic phase. Moreover, the scheme requires certain conditions to work: (1) each good or bad router must be directly connected to at least one good router (2) each good router must be able to send a packet to each other good router through a path of good routers (3) the majority of routers must be good.

1.4.2 Cooperative Anomaly Detection

One of the earliest intrusion detection schemes for ad hoc networks was proposed by Zhang and Lee [24][25]. One of the basic ideas is distributed monitoring and cooperation among all nodes, similar to the basic idea in WATCHERS. Each node independently observes its neighborhood (within its radio range) looking for signs of intrusion. Each node runs an IDS agent which keeps track of internal activities on that node and packet communications within its local neighborhood.

A second idea in the scheme is to rely mainly on anomaly detection because of perceived difficulties with misuse detection. Misuse detection is limited to the set of known attacks with existing signature. Also, signatures must be constantly updated which would be a difficult process in a wireless ad hoc network. Since anamaly detection does not require the distribution of signatures, it is easier to implement in independent nodes. Each node develops a normal profile during a training period, and looks for signficant deviations from the normal profile to detect anomalies.

A third idea in the scheme is cooperation among nodes to cover a broader area. If a node has "strong" evidence of an anomaly, it can raise an alert itself. However, if a node has weak or inconclusive evidence of an anomaly, it can request a global investigation. The requesting node shares its data about the suspected intrusion with its neighboring nodes. The neighboring nodes share their relevant data, and each participating node follows a consensus algorithm to determine whether to raise an alarm. Any node that comes to the conclusion that an intrusion exists can raise an alarm.

The response to an alarm might be recomputation of routing tables to avoid compromised nodes, or communication links between nodes are forced to re-initialize (re-authenticate each other). The latter would not be effective if an attack has compromised a node and captured its authentication credentials.

1.4.3 Watchdogs and Pathraters

The idea of nodes monitoring the packet forwarding behavior of neighboring nodes was also proposed by Marti et al. [26]. Dynamic source routing is assumed. When a packet is ready to be sent, a path to the destination is discovered on demand, and the addresses of the nodes along the path are encapsulated in the packet header. Two new ideas are introduced: watchdogs and pathraters.

A watchdog is a process running on a node to monitor the behavior of neighboring nodes. After a node forwards a packet, the watchdog monitors the next node to see that the packet is forwarded again. With source routing assumed, the watchdog has knowledge of the proper route for a tracked packet. If a neighboring node is observed to drop more packets than a given threshold, that node is deemed to be misbehaving.

The pathrater works to avoid routing packets through misbehaving nodes. Each node maintains a rating for every other node in the range from 0 to 1. It calculates a path metric by averaging the node ratings in the path. Node ratings are initialized to a neutral value of 0.5. Actively used paths are incremented periodically, but nodes suspected of misbehaving will have their rating lowered severely.

Since the watchdog is a rather simple monitoring process, several limitations were noted. First, the scheme is limited to source routing because the watchdog needs knowledge of the proper route for each packet. Second, it is vulnerable to interference by a malicious node falsely reporting other nodes as misbehaving. Third, multiple misbehaving nodes could collectively interfere with the watchdog process. Lastly, a misbehaving node could escape detection by dropping packets just below the threshold level.

1.4.4 TIARA

TIARA (Techniques for Intrusion-resistant Ad Hoc Routing Algorithms) was actually a set of mechanisms to ensure an ad hoc network could continue to operate under hostile adversarial conditions, rather than an intrusion detection scheme [27]. However, a flow monitoring

mechanism in TIARA is designed to detect path failures from misbehaving nodes.

The basic idea is for source nodes to periodically send special "flow status" messages to destination nodes. Flow status messages contain information about the number of packets that have been sent from the source to destination since the previous flow status message. To prevent interference with flow status messages, each message is numbered sequentially (to detect loss) and encrypted with a digital signature (for authentication).

Upon receiving a flow status message, the destination node compares the carried number to the actual number of packets received since the last flow status message. A path failure is notified to the source node if (1) a flow status message has been lost or not received by a specified time interval (2) the actual number of received packets is less than a threshold fraction of the number indicated by the source or (3) the actual number of received packets is much more than the number indicated by the source.

There are two obvious disadvantages of this scheme for intrusion detection. First, a path failure does not identify which specific nodes could be compromised. Second, the flow status messages incur a cost in additional traffic that is proportional to the number of source-destination pairs in the network.

1.4.5 Malcounts

Another distributed intrusion detection system proposed by Bhargava and Agrawal [28] is essentially an enhancement of Zhang and Lee's approach. As before, it is assumed that each node is independently and concurrently monitoring its local neighborhood (nodes within its radio range). AODV (ad hoc on-demand distance vector) routing is assumed. When a packet is ready to be sent, the source node will flood a request through the network; a request successfully reaching the destination will be acknowledged back to the source.

The central idea in the intrusion detection scheme is that each node maintains a "malcount" for neighboring nodes which is the number of observed occurrences of misbehavior. When the malcount for a node exceeds a given threshold, an alert is sent out to other nodes.

The other nodes then check their malcounts for the suspected node and may support the initial alert with secondary alerts. If a suspected node triggers two or more alerts, it is deemed to be malicious and a "purge" message is broadcasted. In response, the suspected node is avoided by the other nodes.

A problem with the proposed scheme is it is not clear if malcounts are only cumulative, so they can increase but not decrease. The ability to decrease malcounts would be useful for nodes with unusual but not malicious behavior that might be falsely identified as malicious. Their unusual behavior might cause their malcount to increase, but then a period of good behavior would result in their malcount returning to a normal value. This could avoid false alerts.

Naturally, this scheme works only if at least two trustworthy nodes is observing a suspected node, and can be defeated by malicious nodes sending out false alerts. Also, the scheme depends on a threshold for malcounts. A compromised node could avoid detection by keeping its misbehavior under the threshold.

1.4.6 CONFIDANT

The CONFIDANT (Cooperation of Nodes: Fairness in Dynamic Ad-hoc Networks) scheme was proposed by Buchegger and Le Boudec [29]. Like previous schemes, it is highly distributed with each node monitoring its local neighborhood and cooperatively sharing information with other nodes. Source routing is assumed so that nodes have knowledge of the correct route for tracked packets. In each node, the CONFIDANT system includes four components: the monitor, reputation system, trust manager, and path manager.

Similar to Zhang and Lee's approach, the monitor in each node observes the activities of neighboring nodes (within radio range) to look for misbehavior. With source routing assumed, the monitor has knowledge of the next hop for each packet. When the node forwards a packet to a neighbor, it watches the neighbor to see whether the packet is forwarded correctly to the next hop. A copy of the entire packet is also stored temporarily to detect any suspicious modifications to the forwarded packet. If a misbehavior is observed, the

reputation system is called.

The reputation system is similar in concept to Bhargava and Agrawal's malcount and Marti et al.'s node ratings. The reputation system consists of a table listing all observed nodes and their reputation ratings. If a node is observed to be misbehaving (deviating from expected routing behavior), the node's rating is changed by a weighting function depending on the confidence in the accuracy of the new observation. To reduce the chance of false alarms, a node's rating can be improved after a specified period of good behavior. If a node's rating falls below a threshold, the path manager is called.

The path manager has a number of responsibilities. It keeps track of a security rating for paths depending on the reputations of nodes in the path. Paths containing a malicious node are deleted. If a received packet is going on a path containing a malicious node, the packet is ignored and the source is alerted. If a received packet comes from a malicious node, the packet is ignored.

The last component, the trust manager is responsible for receiving and sending "alarm" messages. Alarm messages contain information about observed misbehaviors to warn about suspected nodes. Alarm messages are sent to other nodes on a "friends" list although the maintenance of the friends list has not been described. When a node receives an alarm message, the trust manager looks up the source of the message. If the source is trusted, the alarm message is added to a table of alarms. If there is enough evidence that a reported node is indeed malicious, the information is passed to the reputation manager.

A number of details in the CONFIDANT scheme remain to be developed. For example, misbehaviors besides incorrect packet forwarding are not yet specified. Other missing details are the values for thresholds, timeout for improving reputations, and who qualifies for the friends list. Also, the scheme is currently limited to source routing.

1.4.7 MobIDS

MobIDS (Mobile Intrusion Detection System) proposed by Kargl et al. [30] is generally similar to the previous distributed IDS schemes. Multiple sensors in the network keep track of observed instances of cooperative and non-cooperative behavior of nodes. Cooperative instances are given positive values whereas non-cooperative instances are given negative values. All instances observed for a suspected node are combined to calculate a sensor rating for that node, where older instances are given less weight. Then all sensor ratings for a suspected node are averaged, with a weight reflecting the credibility of each sensor, into a "local rating" for that node.

The local ratings are distributed periodically by broadcasting them to neighboring nodes within a given range. Each node averages the local ratings that it receives (including its own rating) into global ratings for other nodes. But global ratings are accepted only when at least a prespecified minimum number of nodes have contributed to the rating. Nodes are deemed to be misbehaving if their ratings drop below a given threshold. Routes are changed to avoid misbehaving nodes, and packets related to those nodes are dropped.

1.4.8 Mobile Agents

Puttini et al. [31] propose a distributed IDS scheme that is similar architecturally to previous proposals except that mobile agents are used for interactions between nodes (instead of data). Mobile agents are software programs that can autonomously suspend execution at one node, transfer their code and state to another node, and resume execution at the second node. Mobile agents are usually implemented in Java because the Java Virtual Machine is widely supported on a broad variety of operating systems.

Each node independently runs a process called local IDS (LIDS). The LIDS includes a sensor that is essentially an SNMP (simple network management protocol) agent to retrieve data from the node's MIB (management information base). The LIDS includes a file of signatures and performs misuse detection to detect attacks.

Information is shared among nodes by dispatching mobile agents, although implementation details about this procedure is lacking. Also, the performance and costs of the mobile agents have not been evaluated. Mobile agents have been studied for many years and proposed for fields such as network management and electronic commerce. However, the theoretical advantages of mobile agents have been elusive.

Mobile agents have never seen much commercial success. Part of the reason is the need for universal adoption of a mobile agent platform (e.g., Java Virtual Machine) which supports the execution and migration of mobile agents. Another reason is that mobile agents do not seem to perform any applications that static agents can not. Finally, mobile agents introduce additional security concerns because they involve the installation of new (possibly untrusted) code on a host. Special security mechanisms must be installed on hosts to ensure that mobile agents do not cause damage. Since mobile agents require higher security, mobile agents are probably poor choices as a solution to security problems such as intrusion detection.

1.4.9 AODVSTAT

AODVSTAT is an extension of STAT (state transision analysis technique) to intrusion detection in wireless networks that use AODV (ad hoc on-demand distance vector) routing [32]. STAT is a stateful signature-based detection technique proposed earlier for wired networks [33]. The premise is that computer attacks can be characterized as sequences of actions taken by an intruder. States represent a snapshot of a host's volative, semi-permanent, and permanent memory.

A complete representation of a successful attack starts from a safe initial state, proceeds through a number of intermediate states, and ends in a compromised state. States are characterized by assertions which are functions with arguments returning boolean values. These assertions describe aspects of the security state of the system. Transitions between states are associated with signature actions, which are actions by the intruder that are necessary for a successful attack. Omission of a signature action would prevent successful completion of the attack.

AODVSTAT applies the ideas of STAT to AODV-routed wireless networks. As mentioned earlier, AODV discovers routes on demand when a packet is ready to be sent. The source node floods a request through the network, and a reply is returned by the destination or an intermediate node that has a route to the destination. A malicious node could interfere with the control packets of the routing protocol, or interfere with the forwarding of data packets.

AODVSTAT sensors are placed on a subset of nodes for promiscuous sensing of radio channels. A sensor has two modes of operation. In stand-alone mode, a sensor looks for signs of attack only within its local neighborhood. In distributed mode, sensors periodically exchange "update" packets containing information about the neighboring nodes of each sensor. The purpose for sharing information is to detect attacks in a distributed way.

As in STAT, AODVSTAT works by stateful signature-based analysis of the observed traffic. Each sensor has a file of attack signatures and looks for a signature match with the traffic. A match triggers a response, usually an alert.

AODVSTAT would have largely the same strengths and weaknesses as STAT. As a misuse detection technique, AODVSTAT could accurately detect types of attacks that consist of sequential actions. A practical issue of how to update the attack signature files at all sensors in an ad hoc network has not been addressed. Also, AODVSTAT has the same limitations as all misuse detection techniques, the inability to detect attacks without an existing signature. However, in a real implementation, it should be straightforward to combine AODVSTAT with anomaly detection for the best of both techniques.

1.4.10 Trust Model

Pirzada and McDonald [34] described an approach to building trust relationships between nodes in an ad hoc network but the method is essentially intrusion detection. It is assumed that nodes in the network passively monitor the packets received and forwarded by other nodes, called events. Events are observed and given a weight depending on the type of application requiring a trust relationship with other nodes. The weights reflect the significance of the observed event to the application. The trust values for all events from a node are

combined using weights to compute an aggregate trust level for another node.

Trust values could be viewed as link weights for the computation of routes. Links with smaller weights would be links to more trusted nodes. A shortest-path routing algorithm would compute the most trustworthy paths.

The similarities between this scheme and previous IDS schemes are clear. Both approaches involve nodes observing the behavior of other nodes and making independent judgments about them. The only difference is that intrusion detection attempts to decide whether a node has been compromised (misbehaving) or not, whereas Pirzada and McDonald's trust model decides on the trustworthiness of a node.

1.4.11 RESANE

RESANE (REputation based Security in Ad hoc NEtworks) [35] takes a view similar to Pirzada and McDonald's trust model. RESANE is not an IDS scheme per se, but uses intrusion detection techniques for a trust model. It assumes that nodes are running an IDS scheme to identify nodes that are misbehaving. The problem addressed is how to make use of the IDS information.

The goal of RESANE is to calculate reputations for nodes and leverage reputations to motivate cooperation between nodes and good behavior throughout the network. The idea is that a bad reputation will motivate a node towards good behavior. If the node continues misbehavior, it reputation will continue to suffer and the node will become isolated from the rest of the network.

A node calculates a reputation rating for a suspected neighbor from the neighbor's misbehaviors observed by the node. The node can also gather reputation ratings for that suspected neighbor from other neighboring nodes that have observed it. If a node detects a misbehavior by a suspected neighbor, the node can proactively broadcast its information to other neighbors to help them protect themselves. Thus, the overall network is protected by cooperative information sharing.

1.4.12 Critical Nodes

Karygiannis et al. advocated the concept of critical nodes [36]. These critical nodes are worth monitoring at the expense of more resources because they have considerable effect on network performance. In other words, if a critical node is malicious or misbehaving or fails, it would significantly degrade network performance. Non-critical nodes are not as important to monitor when resources are limited (the usual case in ad hoc networks).

The notion of critical nodes may aid the problem of intrusion detection, but the work does not address specifically how intrusions may be detected.

1.4.13 SCAN

SCAN attempts to address two problems simultaneously: routing misbehavior (control plane) and packet forwarding misbehavior (data plane) [37]. Routing misbehavior is exhibited by a node that does not participate properly in the routing protocol, e.g., false route advertisements. Packet forwarding misbehavior refers to any intentional interference with the proper relaying of packets, e.g., packet dropping, packet misrouting.

SCAN is based on two central ideas that are similar to previous IDS schemes. First, each node monitors its neighbors independently. Different from a watchdog which looks only for packet forwarding misbehavior, nodes in SCAN observe their neighbors for both routing misbehavior and packet forwarding misbehavior. The second idea is information cross validation. Each node monitors its neighbors by cross checking the overhead transmissions with other nodes. Nodes in a neighborhood collaborate with each other through a distributed consensus protocol. A suspected node can be eventually convicted of being malicious only after multiple neighbors have reached that consensus. This assumes that the network density is sufficiently high that a node can promiscuously overhear the packets sent and received by its neighbors, and nodes have multiple neighbors within range.

For routing misbehavior, SCAN requires two modifications to the usual AODV routing protocol. The usual routing update messages do not contain enough information for nodes

to make judgments about routing misbehavior. First, an additional field for "previous hop" is needed in route request messages. Second, an additional field for "next hop" is needed in route reply messages. This additional information in routing messages allows nodes to maintain part of the routing tables of its neighbors. The redundant routing information enables a node to examine the trustworthiness of future routing updates from its neighbors.

The distributed consensus protocol is based on an "m out of N" algorithm, where N neighbors have been independently observing a suspected node. The suspected node is convicted as malicious if at least m out of the N nodes votes for that decision (based on observed misbehaviors). Various strategies for choosing the value of m as a function of N are proposed: a fixed fraction of N, a constant value k, or a value depending on a probability of correct detection and probability of false alarm.

If a node is convicted of being malicious, it is blocked from access to the network. In SCAN, each node must present a valid token in order to interact with other nodes. Tokens for convicted nodes are revoked, and revoked tokens are tracked by each node by means of a token revocation list. Asymmetric cryptography is used to prevent forged tokens. Each token is signed by the same secret key so it can be verified by a system-wide public key known to all nodes. Tokens are issued and renewed by a distributed algorithm. A token can be signed by a group of collaborating nodes but not by a single node. A token possessed by a node can be renewed by its neighbors if it expires.

SCAN has limitations and involves some overhead in terms of communications and memory. The current SCAN scheme is limited to AODV, but may be extended to other routing protocols if they are appropriately modified (just as AODV messages must be modified with additional fields). Another limitation of SCAN is a requirement for a dense ad hoc network because multiple neighbors must collaborate to form a consensual judgment about a suspected node. Lastly, there is a requirement that collusion among attackers is limited.

1.4.14 Dempster-Shafer

Chen and Venkataramanan [38] addressed the specific problem of combining the observations of multiple neighbors to form a consensual judgment about a suspected node. Dempster-Shafer evidence theory [39] is proposed to be better than simple majority voting or a Bayesian approach. Essentially, Dempster-Shafer theory allows observers to specify a level of uncertainty in their observation. In the context of intrusion detection, if each node has a reputation or trustworthiness rating, that will be reflected by weighting their vote with a corresponding level of uncertainty. In other words, the votes from untrusted nodes will be discounted in comparison with votes from trusted nodes, in forming a consensual judgment.

1.4.15 Optimization of Limited Resources

In wireless networks, nodes may have limited resources to spend on intrusion monitoring and detection. On the other hand, intrusion detection is more effective when more traffic is monitored. The selection of nodes to operate IDS should consider the trade-off between detection efficiency and usage of limited resources. This trade-off was formulated as an integer linear problem, where detection efficiency is maximized subject to a set of resource constraints [40].

The authors also considered a related problem where sensors could be unreliable due to faults, power savings, or compromise [41]. Again, the problem was formulated as an integer linear problem to minimize resource consumption subject to keeping a desired detection probability and the possibility that sensors could be inactive.

1.5 Open Research Issues

For reasons mentioned earlier, intrusion detection is more difficult in wireless mesh networks than wired networks. Intrusion detection continues to be a difficult and open problem even in wired networks. In wired networks, it is relatively easy to collect traffic data, but the

main challenge is detection accuracy. Neither of the two current analysis approaches, misuse detection or anomaly detection, is perfect. Fundamentally, misuse detection needs an attack signature to recognize an attack. New attacks without an existing signature will be missed, resulting in a high rate of false negatives. Also, it takes significant time to develop and distribute a new signature for a new attack. A new attack has a window of opportunity after its first detection where IDSs have not received a new signature yet. A new attack will not be recognizable in the window of opportunity. Anomaly detection has a different challenge: how to construct a normal behavior profile that will yield a low rate of false positives. Detection accuracy will continue to be the main research issue in wireless mesh networks.

1.5.1 Lack of Experience with Wireless Mesh Networks

Another open issue is the lack of experience with incidents in wireless mesh networks. In contrast, security incidents have been occurring in the Internet over the past 30 years. Although no comprehensive database of attacks exists, 30 years of experience have yielded a wealth of information about Internet-based attacks. This wealth of information has helped the Internet security industry grow to considerable size, and a broad range of security products are available.

On the other hand, wireless mesh networks are a recent development, and there is little real experience with security incidents. Attacks are mostly conjectured and theoretical at this point in time. Hence, it is really unknown how to measure the progress or success of research. More real experience is needed but will not be obtainable until wireless mesh networks are deployed more widely in the field.

1.5.2 Evaluation Difficulties

Different IDSs will detect and miss different attacks. A long standing problem has been how to fairly evaluate and compare different IDS. In the past, experiments for wired networks have used test sets of various attacks and measured the detection rate. However, the results will obviously depend on the types of attacks in the test set because different IDS methods

will have different strengths and weaknesses. Experimental comparisons of IDSs may always be controversial. Also considering the lack of experience with real wireless mesh networks, it is difficult to know what types of attacks will be important or realistic.

1.5.3 Intrusion Tolerance

An indirectly related issue is the concept of intrusion tolerance. Intrusion detection attempts to discover the occurrence of attacks and mostly leaves the response to system administrators. Intrusion tolerance recognizes that attacks are inevitable and some attacks will be successful. The idea is to design networks from the beginning to maintain robust operation even in the face of adversarial actions. For example, redundant paths can guarantee that packets will still be delivered if an attacker brings down nodes. Clearly, intrusion tolerance is related to fault tolerance, except that fault tolerance assumes that faults are random and caused by equipment failures. Intrusion tolerance assumes an intelligent attacker capable of strategic actions. Intrusion tolerance for wireless mesh networks is virtually unexplored.

1.6 Conclusion

This chapter has reviewed the basic concepts of intrusion detection and surveyed a number of proposals for intrusion detection in wireless mesh networks. The proposals are mostly for MANETs because wireless mesh networks are a relatively recent development, but the intrusion detection schemes are directly relevant to wireless mesh networks.

A common theme in the research is the notion that nodes should independently and concurrently monitor their local neighborhoods. This is a necessity due to the decentralized nature of wireless mesh networks. A second common theme is the combination of observations from multiple nodes to form a consensual judgment about a suspected node. With these common themes, the various proposed intrusion detection schemes differ mainly in their details and not in their ideas.

At this point, a number of things are clear about the future of intrusion detection. First, there is much room for improvement. The primary measure of effective intrusion detection is low false positives and false negatives. This "proof" has not been convincingly offered by any scheme so far. Second, the challenges imposed by wireless mesh networks imply that the intrusion detection problem will continue to be open for the foreseeable future. Finally, breakthrough progress may not be expected until wireless mesh networks are deployed more widely in the field. At this time, attacks and therefore intrusion detection are largely speculative and theoretical. More real experience with wireless mesh networks will certainly help to catalyze research progress.

References

[1] S. McClure, J. Scambray, G. Kurtz, "Hacking Exposed," 3rd ed., McGraw-Hill, 2001.

[2] L. Zhou, Z. Haas, "Securing ad hoc networks," IEEE Network, vol. 13, Nov./Dec. 1999, pp. pp. 24-30.

[3] H. Deng, W. Li, D. Agrawal, "Routing security in wireless ad hoc networks,"' IEEE Communications Mag., vol. 40, Oct. 2002, pp. 70-75.

[4] K. Sanzgiri, et al., "Authenticated routing for ad hoc networks," IEEE J. on Sel. Areas in Commun., vol. 23, March 2005, pp. 598-610.

[5] N. Salem, J-P. Hubaux, "Securing wireless mesh networks," IEEE Wireless Communications, volume 13, April 2006, pp. 50-55.

[6] C. Basile, Z. Kalbarczyk, R. Iyer, "Neutralization of errors and attacks in wireless ad hoc networks," Int. Conf. on Dependable Systems and Networks (DSN), 2005, pp. 518-527.

[7] N. Milanovic, M. Malek, A. Davidson, V. Milutinovic, "Routing and security in mobile ad hoc networks," Computer, vol. 37, Feb. 2004, pp. 61-65.

[8] H. Yang, et al., "Security in mobile ad hoc networks: challenges and solutions," IEEE Wireless Communications, vol. 11, Feb. 2004, pp. 38-47.

[9] Y-C. Hu, A. Perrig, D. Johnson, "Wormhole attacks in wireless networks," IEEE J. on Sel. Areas in Communications, vol. 24, Feb. 2006, pp. 370-380.

[10] J. McHugh, "Intrusion and intrusion detection", Int. J. of Information Security, volume 1, Aug. 2001, pp. 14-35.

[11] S. Axelsson, "Intrusion detection systems: a survey and taxonomy," Technical Report 99-15, Department of Computer Engineering, Chalmers University of Technology, Sweden, March 2000.

[12] R. Bace, "Intrusion Detection," MacMillan Technical Publishing, 2000.

[13] D. Marchette, "Computer Intrusion Detection and Network Monitoring: A Statistical Viewpoint," Springer-Verlag, 2001.

[14] R. Bejtlich, "The Tao of Network Security Monitoring: Beyond Intrusion Detection," Addison-Wesley, 2005.

[15] S. Northcutt, J. Novak, "Network Intrusion Detection," 3rd ed., Pearson Education, 2003.

[16] S. Northcutt, M. Cooper, M. Fearnow, K. Frederick, "Intrusion Signatures and Analysis," New Riders Publishing, 2001.

[17] K. Cox, C. Gerg, "Snort and IDS Tools," O'Reilly Media, 2004.

[18] R. Bruno, M. Conti, E. Gregori, "Mesh networks: commodity multihop ad hoc networks," IEEE Communications Mag., vol. 43, March 2005, pp. 123-131.

[19] M. Lee, J. Zheng, Y-G. Ko, D. Shrestha, "Emerging standards for wireless mesh networks," IEEE Wireless Communications, vol. 13, April 2006, pp. 56-63.

[20] I. Akyildiz, X. Wang, W. Wang, "Wireless mesh networks: a survey," Computer Networks, volume 47, 2005, pp. 445-487.

[21] I. Akyildiz, X. Wang, "A survey on wireless mesh networks," IEEE Communications Mag., volume 43, Sept. 2005, pp. S23-S30.

[22] A. Mishra, K. Nadkarni, and A. Patcha, "Intrusion detection in wireless ad hoc networks," IEEE Wireless Communications, vol. 11, Feb. 2004, pp. 48-60.

[23] K. Bradley, et al., "Detecting disruptive routers: a distributed network monitoring approach," IEEE Network, vol. 12, Sept./Oct. 1998, pp. 50-60.

[24] Y. Zhang, W. Lee, "Intrusion detection in wireless ad-hoc networks," 6th Annual ACM Int. Conf. on Mobile Computing and Networking, Boston, 2000, pp. 275-283.

[25] Y. Zhang, W. Lee, Y-A. Huang, "Intrusion detection techniques for mobile wireless networks," Wireless Networks, vol. 9, 2003, pp. 545-556.

[26] S. Marti, T. Giuli, K. Lai, M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," 6th Annual ACM Int. Conf. on Mobile Computing and Networking, Boston,

2000, pp. 255-265.

[27] R. Ramanujan, A. Ahamad, J. Bonney, R. Hagelstrom, K. Thurber, "Techniques for intrusion-resistant ad hoc routing algorithms (TIARA)," IEEE MILCOM 2000, Los Angeles, 2000, pp. 660-664.

[28] S. Bhargava, D. Agrawal, "Security enhancements in AODV protocol for wireless ad hoc networks," 2001 IEEE Vehicular Technology Conf. (VTC 2001), 2001, pp. 2143-2147.

[29] S. Buchegger, J-Y. Le Boudec, "Performance analysis of the CONFIDANT protocol (Cooperation of Nodes: Fairness in Dynamic Ad-hoc Networks)," 3rd ACM Int. Symp. on Mobile Ad Hoc Networks and Computing, Switzerland, 2002, pp. 226-236.

[30] F. Kargl, A. Klenk, M. Weber, S. Schlott, "Sensors for detection of misbehaving nodes in MANETs," Detection of Intrusion and Malware and Vulnerability Assessment (DIMVA 2004), Dortmund, Germany, 2004.

[31] R. Puttini, J-M. Percher, L. Me, R. de Sousa, "A fully distributed IDS for MANET," 9th Int. Symp. on Computers and Commun. (ISCC 2004), 2004, pp. 331-338.

[32] G. Vigna, et al., "An intrustion detection tool for AODV-based ad hoc wireless networks," Annual Computer Security Applications Conf. (ACSAC 2004), Tuscon, 2004, pp. 16-27.

[33] K. Ilgun, R. Kemmerer, P. Porras, "State transition analysis: a rule-based intrusion detection approach," IEEE Trans. on Software Engineering, vol. 21, 1995, pp. 181-199.

[34] A. Pirzada, C. McDonald, "Establishing trust in pure ad-hoc networks," 27th Australian Conf. on Computer Science, Dunedin, New Zealand, 2004, pp. 47-54.

[35] Y. Rebahi, V. Mujica, D. Sisalem, "A reputation-based trust mechanism for ad hoc networks," 10th IEEE Symp. on Computers and Communications (ISCC 2005), 2005, pp. 37-42.

[36] A. Karygiannis, E. Antonakakis, A. Apostolopoulos, "Detecting critical nodes for MANET intrusion detection," 2nd Int. Workshop on Security, Privacy, and Trust in Pervasive and Ubiquitous Computing (SecPerU 2006), 2006, pp. 7-15.

[37] H. Yang, J. Shu, X. Meng, S. Lu, "SCAN: self-organized network-layer security in mobile ad hoc networks," IEEE J. on Sel. Areas in Communications, vol. 24, Feb. 2006, pp. 261-273.

[38] T. Chen, V. Venkataramanan, "Dempster-Shafer theory for intrusion detection in ad

hoc networks," IEEE Internet Computing, vol. 9, Nov./Dec. 2005, pp. 35-41.

[39] G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, NJ, 1976.

[40] D. Subhadrabandhu, S. Sarkar, F. Anjum, "A framework for misuse detection in ad hoc networks - part I," IEEE J. on Sel. Areas in Communications, vol. 24, Feb. 2006, pp. 274-289.

[41] D. Subhadrabandhu, S. Sarkar, F. Anjum, "A framework for misuse detection in ad hoc networks - part II," IEEE J. on Sel. Areas in Communications, vol. 24, Feb. 2006, pp. 290-304.

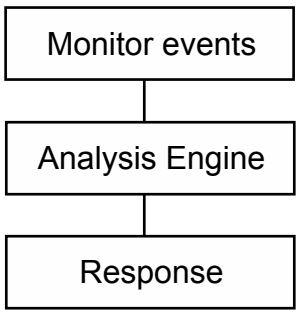

Figure 1.1: Functions of IDS

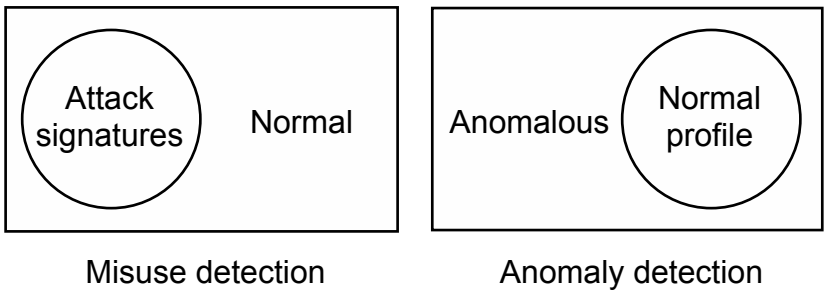

Figure 1.2: Misuse detection and anomaly detection