Contents

Preface...............................................................................................................................................ix
Acknowledgments.............................................................................................................................xi
Editor ............................................................................................................................................. xiii
Contributors .....................................................................................................................................xv

SECTION I Network Traffic Analysis and Management for IDS

Chapter 1 Outlier Detection ................................................................................................................3
Mohiuddin Ahmed, Abdun Naser Mahmood, and Jiankun Hu

Chapter 2 Network Traffic Monitoring and Analysis ........................................................................23
Jeferson Wilian de Godoy Stênico and Lee Luan Ling

Chapter 3 Using Routers and Honeypots in Combination for Collecting Internet Worm Attacks ........................................................................................................................ 47
Mohssen Mohammed and Al-Sakib Khan Pathan

Chapter 4 Attack Severity–Based Honeynet Management Framework ...........................................85
Asit More and Shashikala Tapaswi

SECTION II IDS Issues for Different Infrastructures

Chapter 5 Intrusion Detection Systems for Critical Infrastructure ................................................. 115
Bernardi Pranggono, Kieran McLaughlin, Yi Yang, and Sakir Sezer

Chapter 6 Cyber Security of Smart Grid Infrastructure .................................................................. 139
Adnan Anwar and Abdun Naser Mahmood

Chapter 7 Intrusion Detection and Prevention in Cyber Physical Systems ................................... 155
Mohamed Azab and Mohamed Eltoweissy

Chapter 8 Encrypted Ranked Proximity and Phrase Searching in the Cloud .................................... 187
Steven Zittrower and Cliff C. Zou
Chapter 9  Intrusion Detection for SCADA Systems ................................................................. 211
  Alaa Atassi, Imad H. Elhajj, Ali Chehab, and Ayman Kayssi

Chapter 10  Hardware Techniques for High-Performance Network Intrusion Detection ..........233
  Weirong Jiang and Viktor K. Prasanna

SECTION III  Artificial Intelligence Techniques for IDS

Chapter 11  New Unknown Attack Detection with the Neural Network–Based IDS .............259
  Przemysław Kukiełka and Zbigniew Kotulska

Chapter 12  Artificial Intelligence–Based Intrusion Detection Techniques ..........................285
  Zahra Jadidi, Vallipuram Muthukumarasamy, and Elankayer Sithirasenan

Chapter 13  Applications of Machine Learning in Intrusion Detection ..............................311
  Yuxin Meng, Yang Xiang, and Lam-For Kwok

SECTION IV  IDS for Wireless Systems

Chapter 14  Introduction to Wireless Intrusion Detection Systems ....................................335
  Jonny Milliken

Chapter 15  Cross Layer–Based Intrusion Detection Techniques in Wireless Networks: 
  A Survey .......................................................................................................................... 361
  Subir Halder and Amrita Ghosal

Chapter 16  Intrusion Detection System Architecture for Wireless Sensor Network ............391
  Mohammad Saiful Islam Mamun

Chapter 17  Unique Challenges in WiFi Intrusion Detection .............................................407
  Jonny Milliken

Chapter 18  Intrusion Detection Systems for (Wireless) Automation Systems ..................431
  Jana Krimmling and Peter Langendörfer

Chapter 19  An Innovative Approach of Blending Security Features in Energy-Efficient 
  Routing for a Crowded Network of Wireless Sensors ............................................ 449
  Al-Sakib Khan Pathan and Tarem Ahmed

Index ........................................................................................................................................ 473
Preface

INTRODUCTION

Most of the security threats in various communications networks are posed by the illegitimate entities that enter or intrude within the network perimeter, which could commonly be termed as intruders. Sometimes a legitimate entity in a system could also be compromised in some way so that an attacker-intended task could be performed for breaching the security of the system. To tackle intrusions of various kinds, we commonly hear about intrusion detection systems (IDSs) and intrusion prevention systems (IPSs) or a combination of both called IDPS (intrusion detection and prevention systems). The main task of an IDS is to defend a computer system or computer network by detecting an attack and possibly repelling it. Successful detection of hostile attacks depends on the number and type of appropriate actions. On the other hand, intrusion prevention requires a well-selected combination of baiting and trapping aimed at the investigations of threats. Diverting the intruder’s attention from protected resources is another task. Both the real system and a possible trap system are constantly monitored. Various tasks and functionalities can be thought of under intrusion-related topics in computer, communications, or networking fields:

- Regular checking of the data in computers and systems
- Monitoring and analyzing network traffic
- Analyzing network configuration and vulnerabilities
- Assessing network and data integrity
- Ability to recognize patterns typical to attacks
- Tracking the network policy violations
- Analysis of abnormal activities
- Outside influence and its impact on a system’s security

OBJECTIVE OF THE BOOK

This book compiles the latest trends and issues in intrusion tackling in computer networks and systems, especially in communications networks. It is written for graduate students in universities, researchers, academics, and industry practitioners working in the areas of wired or wireless networking or computer systems, who want to improve their understanding of the interrelated topics.

ABOUT TARGET AUDIENCE AND CONTENT

The target audience of this book is composed of students, professionals, and researchers working in the field of computer and network security especially. Moreover, the book includes some chapters written in a tutorial style so that general readers can be able to easily grasp some of the ideas in the relevant areas. There are a total of four sections of the book with a total of 19 chapters. These chapters have been contributed by authors from 12 countries.

SECTION I: NETWORK TRAFFIC ANALYSIS AND MANAGEMENT FOR IDS

- Chapter 1 - Outlier Detection
- Chapter 2 - Network Traffic Monitoring and Analysis
Preface

• Chapter 3 - Using Routers and Honeypots in Combination for Collecting Internet Worm Attacks
• Chapter 4 - Attack Severity–Based Honeynet Management Framework

SECTION II: IDS ISSUES FOR DIFFERENT INFRASTRUCTURES

• Chapter 5 - Intrusion Detection Systems for Critical Infrastructure
• Chapter 6 - Cyber Security of Smart Grid Infrastructure
• Chapter 7 - Intrusion Detection and Prevention in Cyber Physical Systems
• Chapter 8 - Encrypted Ranked Proximity and Phrase Searching in the Cloud
• Chapter 9 - Intrusion Detection for SCADA Systems
• Chapter 10 - Hardware Techniques for High-Performance Network Intrusion Detection

SECTION III: ARTIFICIAL INTELLIGENCE TECHNIQUES FOR IDS

• Chapter 11 - New Unknown Attack Detection with the Neural Network–Based IDS
• Chapter 12 - Artificial Intelligence–Based Intrusion Detection Techniques
• Chapter 13 - Applications of Machine Learning in Intrusion Detection

SECTION IV: IDS FOR WIRELESS SYSTEMS

• Chapter 14 - An Introduction to Wireless Intrusion Detection Systems
• Chapter 15 - Cross Layer–Based Intrusion Detection Techniques in Wireless Networks: A Survey
• Chapter 16 - Intrusion Detection System Architecture for Wireless Sensor Network
• Chapter 17 - Unique Challenges in WiFi Intrusion Detection
• Chapter 18 - Intrusion Detection Systems for (Wireless) Automation Systems
• Chapter 19 - An Innovative Approach of Blending Security Features in Energy-Efficient Routing for a Crowded Network of Wireless Sensors

The first section contains four chapters, which deal with traffic analysis and management for intrusion detection systems. Concepts such as honeypots, honeynets, network traffic analysis, and the basics of outlier detection are discussed in these chapters. The second section has six chapters that discuss different kinds of IDSs for different infrastructures. The chapters in this section also cover new and emerging technologies and systems such as smart grids, cyber physical systems (CPSs), cloud computing, hardware techniques for high performance intrusion detection, and so on. The third section, with three chapters, is dedicated to artificial intelligence (AI)–related intrusion detection techniques. The fourth section covers intrusion tackling mechanisms for various wireless systems and networks such as wireless sensor networks (WSNs), WiFi (wireless-fidelity), wireless automation systems, and other wireless systems. This section contains six chapters.

MATLAB® is a registered trademark of The MathWorks, Inc. For product information, please contact:

The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098 USA
Tel: (508) 647-7000
Fax: (508) 647-7001
E-mail: info@mathworks.com
Web: http://www.mathworks.com

Buy The State of the Art in Intrusion Prevention and Detection
http://www.crcpress.com/product/isbn/9781482203516
Acknowledgments

I am very much grateful to the Almighty Allah to allow me the time to complete another work of such kind. The entire process has been lengthy, needing non-stop work, interaction with several people in various ways, and firm determination. I am thankful to all the authors, reviewers, and critics who helped me shape the book in a better way. My hearty thanks should go to my loving wife, Labiba Mahmud, who has supported me all throughout the process. Last but not least, I thank the publisher and publication staff for giving me this opportunity and assisting me to work on this book project.

Best wishes,

Al-Sakib Khan Pathan, PhD
Department of Computer Science
International Islamic University Malaysia, Malaysia
14 Introduction to Wireless Intrusion Detection Systems

Jonny Milliken

CONTENTS

14.1 Wireless Intrusion Detection Systems ............................................................. 336
  14.1.1 The Basic Structure of a Wireless Intrusion Detection System .................. 336
14.2 Threat Identification ....................................................................................... 337
14.3 Architecture ................................................................................................. 338
  14.3.1 Embedded vs. Overlay ............................................................................. 339
    14.3.1.1 Embedded ......................................................................................... 339
    14.3.1.2 Overlay ............................................................................................ 339
  14.3.2 Host, Distributed, and Mobile Architectures ........................................... 340
    14.3.2.1 Host ................................................................................................. 340
    14.3.2.2 Distributed ....................................................................................... 340
    14.3.2.3 Mobile Agents .................................................................................. 341
14.4 Data Collection .............................................................................................. 341
  14.4.1 Data Collection Methods ......................................................................... 342
  14.4.2 Metric Categories .................................................................................... 342
    14.4.2.1 System Logs ...................................................................................... 342
    14.4.2.2 System Calls ..................................................................................... 342
    14.4.2.3 SNMP ............................................................................................... 343
    14.4.2.4 Network Traffic ................................................................................ 343
  14.4.3 Network Metrics in Research .................................................................. 343
14.5 Intrusion Detection ......................................................................................... 344
  14.5.1 Detection Methods ................................................................................... 345
    14.5.1.1 Anomaly Detection .......................................................................... 345
    14.5.1.2 Misuse/Signature Detection ............................................................... 346
    14.5.1.3 Specification Detection ...................................................................... 347
    14.5.1.4 Hybrid ............................................................................................... 347
  14.5.2 Machine Learning ..................................................................................... 347
    14.5.2.1 Supervised ....................................................................................... 348
    14.5.2.2 Unsupervised .................................................................................... 348
    14.5.2.3 Semi-Supervised .............................................................................. 348
  14.5.3 Machine Learning Techniques .................................................................. 349
    14.5.3.1 Neural Networks .............................................................................. 349
    14.5.3.2 Self-Organizing Maps ...................................................................... 349
    14.5.3.3 Bayesian Systems ............................................................................. 350
    14.5.3.4 Markov Model .................................................................................. 350
    14.5.3.5 Support Vector Machines ................................................................. 351
14.6 Alert Correlation ............................................................................................ 351
  14.6.1 Pre-Processing ......................................................................................... 352
Security in IT systems is an increasingly important area of research as users have come to accept that every system connected to the Internet is vulnerable. These vulnerabilities can come from known threats, zero-day attacks, malware, or DoS (denial of service) attacks. Some systems that aim to protect against these attacks include firewalls and antivirus systems. Each of these defenses only covers a fraction of computer security however. Firewalls are barriers and do not inform about activity within the network while antivirus systems only protect against malicious software on hosts. There are many more protocol and network level threats, which these options do not protect against.

The primary means of defense against these kinds of attacks is with an IDS (intrusion detection system). An IDS monitors the network environment and alerts a human operator to the presence of an attack or abnormalities. An IDS is also useful for detecting attacks that are difficult and resource intensive to prevent (such as DoS) but can be mitigated once they have begun. WIDS (wireless intrusion detection systems) are designed to mitigate the risks of attacks in WiFi networks by monitoring traffic broadcast over the wireless medium of a network for suspicious activity.

14.1 WIRED INTRUSION DETECTION SYSTEMS

The rudimentary functions of a WIDS encompass many areas, from data collection and attack detection through to attack reporting to a discerning human or automated response system. Most research work concentrates on one specific area of WIDS performance without consideration of the operation of the whole. This can make it difficult to identify how disparate research investigations and conclusions relate to or impact on each other. Hence, it is useful for researchers to have an appreciation of the whole of the system. This section will outline the operation of a typical WIDS, categorized into six sections:

- Threat identification
- Architecture considerations
- Data collection
- Detection strategy
337

Introduction to Wireless Intrusion Detection Systems

- Correlation method
- Evaluation

The first step in the process is to identify which attacks are threats to the network. There are hundreds of attacks that can be lobbied against a system, and it is critically important to identify which of these are priorities because no IDS exists that can reliably detect all known and zero-day attacks.

Once the attacks are identified, the architecture, topography, and topology of the network must be investigated. This reveals the resources available and the potential placement locations for data monitors. Once these are known, the best way of detecting the attacks using the available resources is described using a series of metrics based on the data collection methods. The metrics in question can come from many sources and operate on many OSI (open systems interconnection) layers.

Once the information sources for attack discovery are known, a detection method must be selected. This is an area of mature research as this is the first stage of active intelligence and decision making in the system. It is possible to have a system based on anomaly, signature, specification, or hybrid detection, trained using any number of machine learning algorithms.

In order to reduce the volume of potential alerts, which can confound a human at the evaluation stage, alert correlation is used to refine the WIDS output. The outcome of the correlation process must then be displayed in a descriptive way to the human operator or automated response system for evaluation.

These six categories are applicable for any IDS although the focus in the descriptions will concentrate on Wireless IDSs. The most important categories in this system are the first and last as they encompass the critical link a WIDS is designed to provide: evaluation if a threat (or attack) has occurred. The four intervening stages should be focused toward ensuring that this link is as reliable as possible. Hence, one of the critical factors to appreciate from the classification outlined here is that all subsequent categories are reliant on the performance of previous sections; see Figure 14.1. Poor choices in the design or performance of lower layers can impact on the performance of the entire system further up the chain, leading to cascading suboptimal design.

14.2 THREAT IDENTIFICATION

Network attacks compromise the integrity, confidentiality, and availability of systems. Before a WIDS can begin protecting a system, it must know what it needs to protect the system against. The nature of the threat will govern the characteristics and success of the defense system. However, when the threat is not explicitly known, approximations must be made. Unfortunately, there is no common, standard way of classifying attacks. As a result, it can be very difficult to determine the critical attacks against networked systems. Many approaches for categorization and identification of attacks have been proposed, all of which try to balance completion with accuracy, in the form of taxonomies (Figure 14.2).

A taxonomy is the study of the means of classification, in this case, the way in which attacks are classified. Taxonomies play an important feedback role in a WIDS too because identification

FIGURE 14.1 Structure of a WIDS.

FIGURE 14.2 Threat identification within WIDS categorization.
of attacks is necessary as a barometer for the success of the system. Many taxonomies have been proposed, but none have yet to satisfy all the criteria set out [1]:

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accepted</td>
<td>Deterministic</td>
</tr>
<tr>
<td>Comprehensible</td>
<td>Mutually exclusive</td>
</tr>
<tr>
<td>Complete</td>
<td>Repeatability</td>
</tr>
<tr>
<td>Have established terminology</td>
<td>Unambiguous</td>
</tr>
<tr>
<td>Have established terminology</td>
<td>Useful</td>
</tr>
</tbody>
</table>

The goal of many taxonomies is to comprehensively encompass all relevant attacks. This is academically desirable but does not necessarily address the most common practical limitation: clarity for those of limited technical backgrounds. As network attacks become increasingly common, it becomes more important for non-academics to be able to act securely and know the threats they are subject to. No key taxonomy has yet arisen that is both sufficiently comprehensive and easy to understand.

The complexity of a taxonomy is generally dictated by the volume of attacks it wishes to cover. Reducing this attack space to a manageable level would assist in presentation and interpretation. A subset of potential attacks is given [2] although it is incomplete:

<table>
<thead>
<tr>
<th>Attack Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>External misuse</td>
<td>By bypassing controls</td>
</tr>
<tr>
<td>Hardware misuse</td>
<td>Eavesdropping</td>
</tr>
<tr>
<td>Masquerading</td>
<td>Interference</td>
</tr>
<tr>
<td>Preparatory abuse</td>
<td>Authorization attacks</td>
</tr>
<tr>
<td>Race condition</td>
<td>MITM (man in the middle)</td>
</tr>
<tr>
<td>Privilege escalation</td>
<td>Social engineering</td>
</tr>
<tr>
<td>Vulnerabilities</td>
<td>Access rights</td>
</tr>
<tr>
<td>Misconfiguration</td>
<td>Worms</td>
</tr>
<tr>
<td></td>
<td>Viruses</td>
</tr>
<tr>
<td></td>
<td>DoS</td>
</tr>
<tr>
<td></td>
<td>Spoofing</td>
</tr>
<tr>
<td></td>
<td>Buffer overflow</td>
</tr>
<tr>
<td></td>
<td>Password attacks</td>
</tr>
<tr>
<td></td>
<td>Communication based</td>
</tr>
</tbody>
</table>

Some notable academic investigations into taxonomies include the following:

- **Purdue University** [3]—Taxonomy based on system logs states that a thesaurus of vulnerability terms is needed to remove confusion.
- **Straub and Widom** [4]—Describes a means of classifying attackers based on their motivation as opposed to the attacks themselves and links these to potential responses.
- **AVOIDIT** [5]—Categorizes attacks based on attack method, attack target, operational impact, information impact, and remediation options. Includes a comparison of [1] and [2].
- **Defense-Centric** [6]—Advocates that taxonomies are more useful if built from the defender’s point of view rather than from the attacker’s.
- **IDS Taxonomy** [7]—Develops a means of comparing the performance of IDSs against each other.
- **Communities** [8]—Investigates the impact on communities from cyber attacks with a separation between the intrusive event and the impact of the event.

In each of these cases, attacks have been defined and named differently based on the authors’ experience. There is no formal method applied that would attempt to ensure that all attacks within a particular technology, protocol, or device have been identified. It is possible to list every possible threat that a device may be susceptible to, but this could create many false alarms at evaluation, which can mask the real dangers. No IDS protects against all existing attacks, so the most critical need to be identified instead [9].

### 14.3 ARCHITECTURE

Once the important threats that can be levied against a vulnerable system have been identified, the next decision concerns where to place the components of the WIDS (Figure 14.3). The architectural structure
can play a huge role in detection ability [10]. This choice is dependent upon the availability of resources, equipment, monitoring points, communication channels, etc. Choices may also be dictated by how distributed the network is. For example, if there are many network locations that need to be protected, then selecting data collection points closer to centralized network switching positions may be prudent.

If an inappropriate architecture is selected, then a WIDS can severely hamper network performance by consuming excessive resources or not having sufficient information available in order to make accurate detection decisions further on in the process. Despite this, very limited research has investigated how to optimally select detection locations.

14.3.1 Embedded vs. Overlay

Depending on the availability of computing resources, a WIDS can be more effectively installed as either an embedded or an overlay system. An overlay monitoring network is a system that is independent from the network that is to be monitored and is designed to collect the same data from the wireless medium but with minimal network disruption [11]. This is in contrast to an embedded system, which connects directly with the equipment that is to be monitored. This often includes routers or switches and WLAN access points. There are positive and negative aspects for each approach to WLAN monitoring, as summarized below.

14.3.1.1 Embedded

The traffic that the WIDS must analyze already passes through the network, requiring protection, so the optimal solution would seem to be an embedded system. This utilizes the spare capacity in bandwidth and processing power from the existing equipment to monitor, track, and communicate intrusion information. There are several benefits to this approach:

- Low detection latency
- Low response latency
- Low or potentially zero equipment cost
- Existing communication channels are used

Nonetheless, there are drawbacks to this implementation. The reason for this is resource, rather than technology, based:

- Infringes on existing network performance
- Crashing the network node crashes the defense
- No redundancy
- May require router upgrades or modification
- Self DoS conditions

Any additional load that is placed on the existing infrastructure is very likely to have a performance trade-off. One solution would be to replace the network components with upgraded systems, but then an embedded solution loses its main cost benefit over overlay solutions. There are security considerations too; if an adversary crashes the operational router, then it crashes the WIDS as well.

14.3.1.2 Overlay

In an overlay system, an entirely new monitoring network is deployed alongside the existing infrastructure. A WIDS constructed in this fashion will attempt to collect a majority of wireless traffic
passing to the AP (access point) under protection directly from the wireless medium. It is not necessary to directly replicate every aspect of the network. Depending on the geography of the area, it may be possible to have one device collect information for many APs. Some positives of this system are the following:

- Equipment diversity
- Multiple source monitoring
- No network performance impact
- Larger resources for monitoring system
- No self DoS

This implementation prioritizes the operational network performance above WIDS performance but requires additional work to plan and install. Given that another entirely separate network is deployed, this can significantly increase the amount of work for security staff. There is also the potential for interference between the operational network and WIDS WiFi channels. Drawbacks include the following:

- Generally larger deployment cost overall
- Slower response and notification times
- Only monitors network traffic
- Increases network administration required
- Potential privacy issues in collecting unintended data

14.3.2 Host, Distributed, and Mobile Architectures

14.3.2.1 Host

Implementation of a host architecture ensures that all components of the IDS remain within the same physical hardware and do not rely on communication between other hosts. This allows intrusion detection within the host itself but not necessarily outside of it [12]. Communication to and from the host can be studied, but there is little or no information sharing between hosts and no means of corroborating data. Attacks that target multiple hosts (such as port scans) cannot be detected.

There are some system-level metrics that can only be gathered by monitoring on the host itself, such as system calls or many specification signatures. It also guarantees that only the host itself is compromised through any attack or compromise of the IDS itself.

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unique detection metrics</td>
<td>Expecting poor/no inter-host communications</td>
</tr>
<tr>
<td>Only consumes host resources</td>
<td>No distributed attack detection</td>
</tr>
<tr>
<td></td>
<td>Poor network/protocol detection</td>
</tr>
<tr>
<td></td>
<td>Attacks on host compromise IDS</td>
</tr>
<tr>
<td></td>
<td>No overlay available</td>
</tr>
</tbody>
</table>

14.3.2.2 Distributed

Distributed systems can operate at many different levels throughout the network hierarchy, in switches, routers, hosts, etc. [13]. Concentrating monitors within network equipment focuses on communication between hosts and alleviates some of the problems of host monitoring; no processing impact on the hosts, observing network events, and isolating monitoring station from host compromise.
There are drawbacks with this system. Because each host on a system may be different, the data for a network monitor can appear conflicting. The volume of data to process can grow exponentially as a deployment increases in size. Also, data that is encrypted will pass through uninspected by the equipment. Any network that utilizes a large number of protocols or applications can confuse or confound a distributed WIDS and the processing required to alleviate this problem can be prohibitive.

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Can detect network/protocol attacks</td>
<td>• Can struggle with disparate data</td>
</tr>
<tr>
<td>• Removes burden from users</td>
<td>• Loads network communication infrastructure</td>
</tr>
<tr>
<td>• Redundancy possible</td>
<td></td>
</tr>
<tr>
<td>• Scales well</td>
<td></td>
</tr>
</tbody>
</table>

### 14.3.2.3 Mobile Agents

Mobile agents cooperatively assist with intrusion detection in a dynamic environment [14]. While the previous systems rely on different placement and communication methods between stationary agents, in a mobile environment the agents themselves can move. Agents can be given specific roles within the network, for example, a monitor agent, analysis agent, retrieval agent, result agent, executive agent and manager agent [15].

Agents allow several independent and intelligent processes to cooperate in securing the network. While this allows distributed computation, asynchronous operation, and an updatable modular structure, there are questions of efficiency and security. The benefit to this approach is that, instead of duplicating agents over every monitoring point, you may instead create a smaller number of agents, which can transfer themselves through the network to provide coverage. These agents can be coordinated by a command structure to investigate suspicious activities. In theory, this should reduce network traffic, add redundancy to the system, and allow more efficient and directed response to intrusions.

There are some problems with this approach [16], such as the following:

- Expensive to design
- Difficult to quantify performance improvements
- Difficult to develop, test, and debug
- Poor security and control
- Can be brainwashed
- Cannot keep secrets
- Lacks necessary infrastructure support

### 14.4 DATA COLLECTION

Once the attacks that are of primary concern to a WIDS are chosen and the available network locations and resources available have been identified, a decision must be made on what data to collect. Attacks may be occurring within the architecture, but if the correct data is not collected from these locations, all subsequent stages in WIDS design can be compromised. The choice of incorrect metrics can restrict the options for detection algorithms, correlation techniques, and response mechanisms (Figure 14.4).

**FIGURE 14.4** Data collection within WIDS categorization.
Unfortunately, at present, there is no standard for selecting, measuring, or tracking metrics. This leads to conflicting measuring approaches and conflicting research outcomes in some instances. As a result, there has been a trend in more recent publications toward identifying and classifying the features or metrics that are optimal [17]. Optimizing and prioritizing metrics is an important goal for WIDS research because it is possible for multiple attacks to be detectable via tracking a single metric while some attacks may require multiple metrics before they can be reliably detected. Optimal selection can reduce false positives, reduce false negatives, and improve root causing of alerts.

14.4.1 Data Collection Methods
An important aspect of data collection is the method of collecting data. For WIDS research, one means of addressing this problem is the identification of appropriate data sets. It is not uncommon for research papers to generate their own data set; however, this presents problems in comparing data sets and the results based on them [18]. Selecting a suitable data source is an important factor in ensuring that the results drawn from experiments are accurate and relevant. This topic is dealt with in greater depth elsewhere in this book (Chapter 17).

14.4.2 Metric Categories
It is not the case that the more metrics monitored, the more secure the system. Some metrics are more useful than others in detecting attacks, and irrelevant metrics may confound the detection system or human response [19]. Sources of metrics can come from all layers of the communication stack, any protocols that operate over the network and, potentially, the information from any system or process in operation. This creates a problem for optimal selection because many metrics may not be available or practical for tracking, depending on the environment the WIDS has been deployed in. Hence, metric possibilities are generally restricted to the most common protocols and network behavior to attempt to create solutions that are likely to work on many different systems installed in many different locations. Common metrics that are used in a WIDS can be separated into four categories:

- System log files
- System calls
- SNMP
- Network packets

14.4.2.1 System Logs
System or audit logs are sets of events created by the OS (operating system) for performing tasks and thus can only be tracked from within host systems [20]. Logs usually represent a list of the running processes on the machine and past activity on the machine. These are typically used with anomaly detection techniques to build applications policy.

The drawback with these logs is that each OS will create, store, and represent them differently, and there is no common format for intrusion detection. Work [21] proposes a specific language to define the meaning of intrusion events. These logs have an associated security risk as well because any attacker that can gain access may well be able to discover more information about the victim than if the monitor was not present.

14.4.2.2 System Calls
System calls are used for tracking illegitimate behavior of a program installed on a protected system. Should a program act maliciously or anomalously, it must communicate with the operating
system. However, this means that the communication can be different across each OS. The process of determining these calls for the huge library of common programs in use is non-trivial and so tends to be performed only for critical programs. It is also possible to subvert the detection scheme in use by wrapping an attack within legitimate system calls [22].

14.4.2.3 **SNMP**

The SNMP (simple network management protocol) allows for various status updates between devices within a network and is routinely used for remote administration of network performance. It has been suggested that combining this information with an intrusion detection system can aid with detection effectiveness [23].

The TCP, IP, and ICMP data [24] are combined with system configuration, network traffic, control, and error statistics from the MIB (SNMP knowledge base). In an experiment in which the detection levels were set to 30%, the results indicate that most attacks can be detected to within a 95% success rate with less than a 1% false positive rate.

14.4.2.4 **Network Traffic**

Analyzing frame data passing through the network can provide information about the security of the network users and the network infrastructure. However, there are restrictions and challenges with inspecting this data in some cases. Due to privacy limitations, much of the data from the network traffic payload may not be accessible in all cases [18]. Hence, information from protocol headers is the primary source of this network traffic information. This can cause problems for overlay WIDS, which may then only be able to rely on WiFi management frames for detection purposes [11]. Network traffic is the primary metric source for WIDS, so a more in-depth analysis of the metrics available is considered here.

14.4.3 **Network Metrics in Research**

Many common metrics for IDSs are resident in the network (OSI Layer 3) and transport (OSI Layer 4) layers. As these protocols operate principally the same on wired and wireless installations, this theoretically allows conclusions to be applied to both. This is generally not tested however. Use of metrics at the physical (OSI Layer 1) and data link (OSI Layer 2) layers would deviate significantly for wired and wireless IDSs. Unfortunately, there are few metric investigations that investigate metrics for WIDS [19,25] and fewer at lower layers for WIDS [26]. Some examples of relevant metric selection investigations are outlined below.

Qu et al. [27] utilized a distance method for establishing how far a metric must deviate from its expected value before it can be considered anomalous. The metrics under consideration include the rate of outgoing TCP SYN packets, total number of outgoing UDP packets, ARP request rate, memory usage, and CPU utilization taken every second. No information on the levels or success of these metrics is given.

Chebrolu et al. [28] identified 12 attributes as key attack detection metrics. The exact relationship between each variable and their detection performance is not given. The metrics are the following:

- Service type
- Source
- Destination
- Logged in
- Packet count
- Error rate
- Service count
- Srv_error
- Srv_diff_host_rate
- Dst_host_count
- Dst_host_srv_count
- Dst_host_diff_srv_rate
Lu and Traore [29] detected DDoS attacks using the ratio of incoming IP traffic to outgoing IP traffic. An outlier removal strategy is employed using a Gaussian mixture model and an expectation maximum algorithm. Using this system, the response time was approximately 35 seconds for selected flood attacks.

Zargar and Kabiri [19] determined that certain metrics are better at detecting particular attacks. Key results include the following:

- SYN flags and stream index were most indicative of a DoS.
- Fragmentation commands best indicated a “user to root” attack.
- Distinction between “remote to local” attacks and normal traffic can be determined by a threshold.
- FIN flags and PUSH flags determine a port scan.

Milliken et al. [26] identified challenges and proposed means for detecting flooding DoS attacks using WiFi management frames in an overlay detection network. The work ascertains that an additional packet reception timeout metric is specifically required by an overlay network in order to function effectively.

### 14.5 INTRUSION DETECTION

Once the priority attacks have been identified, architecture and resources allocated, and data collection points established, the method of detecting intrusions must be chosen. This detection process is usually the most processing intensive as it requires operating potentially sophisticated algorithms over large volumes of data. Should the detection algorithm determine that an attack has occurred, then the alert is passed upward to subsequent stages for correlation and evaluation (Figure 14.5).

Detection algorithm testing and development is the most intensely researched field of the stages in WIDS operation outlined here. Nonetheless, it is heavily reliant on the three stages that precede it. If the data fed to the algorithm is incorrect, then poor performance is guaranteed. However there is little research to date that investigates the performance of an algorithm depending on the quality of the data provided [26].

The main objective of this stage is to differentiate normal traffic from potentially intrusive traffic. There are two major assumptions often used at this stage:

1. Attack traffic is inherently different from normal traffic.
2. Normal traffic is more prevalent than attack traffic.

Particularly in modern governmental and large-scale commercial systems at risk from APT (advanced persistent threat), the second assumption is becoming less applicable. Nonetheless, for the majority of systems, it holds true.

Development of an intrusion detection algorithm can be separated into two distinct fields:

- Detection methods
- Machine learning

**FIGURE 14.5** Detection algorithms within WIDS categorization.
Choice of detection method concerns the means of identifying attack signatures from changes in the chosen metrics. This directly addresses the goal of identifying attack traffic among legitimate traffic. Machine learning approaches the issue of teaching a machine to make these decisions. With the large volume of traffic passing through WIDSs, it is impractical to have a human observe and identify all the trends; computer automation must be employed.

14.5.1 Detection Methods

Choice of detection method is the first component of a WIDS with decision-making intelligence. All previous stages have primarily relied on the work of experienced humans to make decisions. Detection techniques are one of the chief sources of false positives in the system, and the method employed can make a large difference to the eventual WIDS performance. The goal is to strike an effective balance between detection rate and the rate of false positives.

Typically, detection methods employed to differentiate legitimate and malicious traffic can be separated into one of four categories:

- **Anomaly detection**: Generates an idea of the normal traffic characteristics by observing normal network operation and detecting any deviations from this; i.e., anything outside this expected norm is an intrusion.
- **Misuse/signature detection**: Establishes a list of rules that should be not violated or known operations that represent attack heuristics; i.e., anything that matches this pattern is an intrusion.
- **Specification detection**: Categorizes all the normal and illegal operations of processes and determines which of the two the current activity represents. This lies between anomaly and misuse detection on the spectrum; i.e., specific processes that perform non-allowed actions are intrusions.
- **Hybrid detection**: Combines the best parts of any two of these approaches and uses them to offset the drawbacks of other areas. Much research into detection is concerned with how to get combinations of anomaly, misuse, or specification detection to work together effectively.

14.5.1.1 Anomaly Detection

Anomaly detection aims to establish a model for the normal operation of the network. Comparing current traffic to expectedly normal limits should mean that any anomalies will indicate intrusion or suspicious activity. This principle works under the assumption that abnormal traffic is distinct from normal traffic and that it is less common. If the assumptions hold then, in general, this technique is capable of identifying novel attacks because even zero-day attacks should deviate from the expected norm. As with any system however this is an approximation of real life, represented by a finite number of attributes, so the model will always be limited. This limitation can lead to false positives (new traffic that is legitimate) or false negatives (attacks newly disguised as legitimate traffic).

A graphical representation of the typical operation of an anomaly detection algorithm is given in Figure 14.6. Note that the detection structure is made up of the profile, decisions, and responses. Profiles are necessary for each of the networks under protection, so if there are multiple devices or users, then the volume of profile data grows. The decision determining if traffic is statistically deviant is a comparison between incoming traffic and this profile. The attack decision allows for the profile to be adaptable and dynamic; however it may not be employed in all systems. The response level will be passed upward to a correlation engine before evaluation. Two important features that impact the success of this system are the quality of the profile generated and the “statistically deviant” decision technique employed.
14.5.1.2 Misuse/Signature Detection

Misuse detection aims to identify intrusions by matching traffic to specific strings of known attack patterns. This is in contrast with anomaly detection, which tries to identify everything that does not fall within its bounds. Signature detection is the same process as misuse except that the patterns are defined by a human expert rather than computer learning. The technique has proven very effective at detecting known attacks and can give a good root cause explanation for the alert it generates.

Because signatures must be developed from known attacks, this detection method is entirely unable to identify novel intrusions. Furthermore, developing these patterns is a difficult and time-consuming process, whether done by hand or by machine and will always be limited by the inability to perfectly replicate real life, which contributes to false positives in the same fashion as anomaly detection. Patterns tend to be developed from historical attack data, which means that the attacks themselves are used less regularly, which causes the rules to become dated. The approach is also defeated by attacks that use a series of steps that could be innocuous in isolation but in a structured way can be used to compromise the system.

A graphical representation of the typical operation of a misuse/signature detection algorithm is given in Figure 14.7. One of the key influences on the performance, similar to profile generation in anomaly detection, is the generation and quality of the rule set. There are challenges around rules covering multiple occurrences and overlapping [30] because rules are added sequentially, not iteratively. This is not aided by the lack of a standard form and the format of IDS rules across systems [30] advocates an algorithm for determining rule clashes. These clashes can be both between rules or within one rule itself and be based on redundancy, verbosity, inefficiency, duplication, etc.

**FIGURE 14.6** Anomaly detection process flow.

**FIGURE 14.7** Rule/misuse detection process flow.
14.5.1.3 Specification Detection

Specification approaches occupy the middle ground between misuse and anomaly detection. They aim to create a system behavioral specification under the assumption that a legitimate and well-behaved system will only operate within these bounds, and any movement outside this can be considered an intrusion. This is functionally different from anomaly detection as it identifies a list of activities a system may not do, rather than identifying uncommon activities. It is functionally different from misuse/signature detection as it identifies what a system may do, rather than only identifying what it may not.

The limitations of the specification are created through expert knowledge rather than machine learning, which suffers from many of the same challenges from previous approaches, particularly ensuring completeness. Specification detection should be able to detect both known and novel attack approaches; however, it suffers in terms of workload because creating these specifications for the large amount of common programs in use today is certainly not a trivial task. Even in instances in which a machine can generate some specifications, they still need to be verified by a human expert at some stage of the process. Some investigations into the feasibility of this system have been performed [31], in particular by [32] in WLANs.

14.5.1.4 Hybrid

Due to the benefits and drawbacks of each of these systems, it is clear that a combination (usually misuse and anomaly) would provide improved detection results, for example, allowing anomaly detection to handle unknown events while misuse detection identifies known attack signatures [33]. Such an approach should decrease the level of false positives if a sufficient method of managing conflicting decisions from multiple detection approaches can be properly managed. Some approaches have also married two anomaly detection engines together in order to try to balance the false positive rate of one against the other. A graphical representation of the typical operation of a hybrid detection algorithm is given in Figure 14.8.

14.5.2 Machine Learning

One of the major challenges in creating an effective intrusion detection algorithm is the difficulty of developing appropriate rules, profiles, or specifications. These attributes need to be both specific

![Figure 14.8 Hybrid process flow.](http://www.crcpress.com/product/isbn/9781482203516)
enough to identify attacks amongst normal traffic and general enough to apply in many different scenarios, locations, and network environments.

Relying on a human to design detection characteristics is highly reliant on the particular knowledge and beliefs of the human and can vary considerably. Machine learning is an area of research that aims to alleviate this problem either partially or entirely by providing the algorithm with a composition of training data. This data should be based on real-life traffic in that it should be primarily real traffic but can have specific attack instances added in to bias detection ability. Within machine learning for IDS, the machine can be taught to detect attacks within this data in any one of three ways:

- Supervised
- Unsupervised
- Semi-supervised

14.5.2.1 Supervised
Supervised machine learning relies on a human element to train the learning process of the machine so that it can determine which metrics indicate an attack and which indicate normal traffic. In supervised learning, the entirety of the data is labeled as either normal or attack data by a human. The machine uses this data to form thresholds, clusters, states, or relationships for generation of a set or rules or profiles [34].

The benefit of this approach is that it allows the machine to make connections that may be too sophisticated for a human to identify or which a human may erroneously omit. This approach also allows for constant, automatic updating of the detection parameters as more traffic travels through the network. Drawbacks include the remaining need for a human expert to identify the positive and negative traffic, which is a non-trivial task. Each expert may label data differently or suboptimally. The training data set is furthermore unlikely to be able to cover all possible eventualities of the system.

14.5.2.2 Unsupervised
Unsupervised learning relies heavily on the assumption that normal network traffic is appreciatively distinct from and more plentiful than abnormal traffic, and so a machine should be able to distinguish between the two without human guidance. A further assumption is that any large, frequent groups of calls or state transitions are likely to be normal rather than abnormal. If both of these assertions hold true, then a larger, unlabeled training set can be used.

This system does not require human guidance and can theoretically detect novel attacks, rather than being restricted to those attacks that a human is aware of and able to label [35]. It is also more likely to generate comprehensive rules or profiles that cover many eventualities. The drawbacks of this system are numerous. First, the rules and profiles generated may be too complicated for a human to interpret easily. This can make it challenging to provide root causes for detection alerts, which makes attribution and response recommendations more difficult. It also cannot account for traffic or nuances from the real world that a human may contribute.

14.5.2.3 Semi-Supervised
Semi-supervised learning occupies a midway point between supervised and unsupervised learning. In this approach, only the conclusively known or a subset of conclusively known [36] traffic is labeled by a human. This reduces the labeling burden on the human and does not require labeling of complex or distributed attacks, which can be time consuming. This allows the system to create parameters for suspicious or attack activities and can potentially differentiate between different attacks rather than normal and abnormal. Nonetheless, identification of conclusively good traffic is still a difficult task for a human to carry out correctly. Identifying “anomalous” rather than attack traffic has been discussed [37].
14.5.3 Machine Learning Techniques

A large number of techniques have been suggested for machine learning in intrusion detection; some of the more prominent methods are outlined here.

14.5.3.1 Neural Networks

Neural networks consist of interconnected nodes, or neurons, which are used for information processing based on the weighted connections between the nodes. The system can adapt the weighting of the node connections depending on incoming data. It is often demonstrated as a MLP (multi-layer perceptron) as seen in Figure 14.9.

In intrusion detection, the connections between nodes represent probable chances of transitions. The weighting of nodes is trained into establishing what a profile should be for the given system. The neural network is then able to identify behavior outside of this normal bound. As the system operates, the accuracy of the node weightings should increase and be more reactive to detection of abnormal values.

While this approach is well used in intrusion detection [38], it does suffer from the potential to allow anomalous behavior to be classified as legitimate, and attack root causing and attribution are not always clear. Operation and training of neural networks tends to be expensive [28].

14.5.3.2 Self-Organizing Maps

The SOM (self-organizing map) is a neural network model that maps multi-dimensional relationships between parameters into a two dimensional map used to analyze and visualize attack/security topography as described in [33].

Each model is formed of neurons (i) in a lattice or grid, in which each neuron has a number of associated n parameters (weight, reference, codebook, etc.). Adjacent neurons form a neighborhood for (i). A neighborhood function determines how closely related (i) is to its neighbors with the more neighbors giving a more accurate result/generalization.

The key advantage of SOM is the formation of clusters, which helps to reduce the input space into representative features. Hence, the underlying structure is maintained while the dimensionality of the space is reduced. There are some drawbacks, however. For example, SOM uses a fixed architecture in terms of number and arrangement of nodes, which has to be defined before training. For largely unknown input data characteristics, it is challenging to determine the network architecture that yields optimal results. Also, the topology of the input space has to match the topology of the output space that is to be represented. However, in real-world data sets, the output must be defined before learning can begin even though the input dimensions may not yet be known [39].
14.5.3.3 Bayesian Systems
Bayesian networks model probabilistic relationships between variables of interest and are very similar to neural networks. Here, connections represent conditional dependencies, and nodes that are not connected to each other represent variables that are conditionally independent, regularly described as a DAG (directed acyclic graph) [28] as in Figure 14.10. In a DAG, each node represents a domain variable, and each edge between nodes indicates a dependency, usually based on probabilities. Thus the probability of the event occurring is based on the evidence for the event based on the parent nodes (the posterior probability).

If these probabilities are calculated for all states, then an idea of the condition of the system as a whole can be established. However, this theory is based on previously observed distributions for each state and relies on the potentially unreasonable condition that states are independent. The major benefit of Bayesian approaches over the likes of neural networks and decision trees [40] is that they can closely represent the inter-dependent relationships among data attributes. It is also possible to add decision nodes to extend the system into decision analysis. These networks are fast, efficient, adaptive, offer good generalizations, and are quickly trained.

14.5.3.4 Markov Model
Markov chains generate a series of state transitions, which, if violated, flag intrusions. This technique regards events as state variables in a process. An event is considered anomalous if it occurs out of sequence or with a low probability of connection with its previous state [41]. In a first-order Markov chain, the next state depends only on the current state as in Figure 14.11. There are also higher-order Markov chains, in which the probability of the next state depends on some fixed number of previous states [42]. The training stage can evaluate the states in terms of internal movements or, in a hidden Markov model, on the outputs of the system. These are commonly used in IDSs and perform well against behavioral deviations.
14.5.3.5 Support Vector Machines
SVMs (support vector machines) create hyper-plane delimitations based on distances between points, creating maximum segments of classification [43]. The SVM finds the optimal separating plane between members and non-members of a class in a feature space. The margin, as indicated in Figure 14.12, represents the level to which the hyper-plane has managed to separate the classes, which should be maximal. However, SVM is a purely binary system and will only identify the divisions between two groups. It requires a small data sample for training and is not sensitive to the dimension of data. This approach has been shown to be effective for intrusion detection although it is more resource intensive and requires more training time.

Yu et al. [44] propose that a two-tier SVM implementation can provide the best results. The first stage categorizes the traffic into normal and abnormal, and the second stage utilizes a multi-stage SVM to identify the different attacks that are taking place, creating a hierarchy of SVMs.

14.6 ALERT CORRELATION
Once the detection algorithm has analyzed the metrics that have been provided, it generates detection alerts based on the belief that an intrusion has occurred. In a reasonably large WIDS installation, there may be multiple intrusion detection components deployed. A large number of detectors can generate a large number of alerts, potentially based on the same event. Therefore the generated alerts can be complementary, contradictory, true, false, or incomplete. Each of the alerts may also have different priorities or response time constraints. Correlating these alerts can help identify attacks, reduce unimportant events, and improve evaluation and response activities (Figure 14.13).

The ability of a correlation engine to correctly group alerts is directly related to the quality, accuracy, and completeness of data generated at the detection level and at subsequent stages that have contributed to detection performance. The most important difference between the alert correlation and intrusion detection stages is that while detection is concerned with separating “good” traffic from “bad” traffic (a divisive process), correlation is concerned with bringing those alerts with similar features together (a cohesive process). The authors in [45] observe a reduction in alert

FIGURE 14.12 Support vector machine example.

FIGURE 14.13 Correlation method within WIDS categorization.
volumes, using a correlation process in the range of 50%–99% over thousands of alerts. Generally, the steps of alert correlation can be divided into three categories:

- Pre-processing
- Correlation
- Post-processing

### 14.6.1 Pre-Processing

The pre-process step converts alerts from various sources into a normalized format and combines multiple alerts into a single alert, removing duplicates and significantly reducing the amount of time processing and evaluation require.

#### 14.6.1.1 Normalization

This step converts alerts into a generic format and reduces the number of alerts to be correlated. One method for normalizing this data into a useful standard is the IDMEF (intrusion detection message exchange framework). The framework requires alerts to conform to nine different attributes:

\[
\begin{array}{ccc}
    \text{Analyzer} & \text{Detection time} & \text{Target} \\
    \text{Create time} & \text{Analyzer time} & \text{Assessment} \\
    \text{Classification} & \text{Source} & \text{Additional data} \\
\end{array}
\]

#### 14.6.1.2 Data Reduction

Reducing the data in the pre-processing stage removes redundant alerts from the processing chain. This speeds up the system, makes it more accurate and reduces the load on the human or automated response system [46]. Alerts may be

- **Aggregated**—Duplicate alerts coming from the same sensor or from different sensors. Aggregation characteristics include timestamp, source IP, destination IP, port(s), user name, process name, attack class, and sensor ID.
- **Filtered**—Removing low interest alert classes and known false alerts. These alerts are normally predefined by administrators.

Investigations in [47] advocate the use of run length encoding (RLE) to reduce alerts specifically from alert flooding attacks against a WIDS. Invoking RLE during high alert volume instances can greatly cut down on overload on the system. Only specific timing data is sacrificed.

### 14.6.2 Correlation Techniques

Correlation utilizes techniques such as feature similarity, known scenarios, prerequisite and consequence, to establish logical connections between alerts or to identify attacks that occur in stages. As the complexity and volume of attacks increases, the ability of a human or automated response system to derive meaning and context from these alerts decreases. Thus reliance on raw alert data is becoming less and less reasonable in a practical context [48].

#### 14.6.2.1 Feature Similarity

Feature similarity clusters alert based on similarity in parameters, such as source IP, port number, target IP, etc., but cannot determine causal relationships between alerts. Links can be established for parameters, such as frequency of alerts and the number of links or associations between alerts.

For the feature similarity approach in [48], the features for the correlation engine to scrutinize included the following:
• Similarity between source IP
• Similarity between target IP
• Similarity in target port numbers
• Similarity between target IP and subsequent source IP
• Backward correlation
• Frequency of alert correlation

14.6.2.2 Known Attack Scenario

Known attack scenarios are coded using either expert rules or machine learned training rules. This uncovers the causal relationship between alerts but can only detect known intrusions. It fundamentally operates on states and transitions and attempts to identify patterns. Features in use in this approach include the following:

• Alert type (time and duplication)
• Time between alerts
• Similarity of consecutive bit of destination IP
• Similarity of consecutive bit of source IP
• Similarity of consecutive bit of last destination IP vs. new source IP

As is noted in [49] it does not necessarily follow that identified scenarios are actually intrusions. The resulting scenarios give watching administrators a better representation of the actions of the network and the ability to make more informed decisions. By grouping them into scenarios, it is hoped that false alerts will be more readily identified, and the false alert rate will drop.

Each time a new alert is produced, the likelihood that it belongs to an existing scenario is calculated. If it is unique, then a new scenario is constructed. Training on human sanitized data is needed to learn the appropriate probability measures.

14.6.2.3 Prerequisite/Consequence

The principle of this approach is that alerts do not occur in isolation; there is very often a pattern or trail of alerts from attack beginning to execution [50]. Recognizing early signs of attack can help to prevent the more damaging later stages from occurring. Combinations of alerts are generally formed with “fact, precondition, consequence” triplets in which fact is the attribute name, and precondition and consequence are logical combinations of events. The drawbacks with this approach are that it cannot detect unknown attacks, and even for known attacks, the future steps may be unclear or too numerous.

In [51], the authors implement techniques to cope with variations in attack strategy and a method of measuring the similarity between attack signatures. DAGs are automatically extracted from correlated alerts by first aggregating intrusion alerts that belong to the same step of a sequence of attacks and then extracting the constraints between the attack steps. Error-tolerant graph isomorphism is used to establish whether generated graphs are unique, similar, or subsets of each other. Using a distance calculation between graphs, the minimum number of edits necessary to change one DAG into another is the similarity measurement metric. While this approach can be computationally expensive if the graphs are large, the authors assume that in reality attack graphs will be small.

14.6.3 Post-Processing

The post-process step is used as a feedback mechanism to improve the performance of pre-processing and correlation, ranking, and prioritizing processed alerts. There may also be an intention recognition function, with which the system infers the end goal of any successful attack, informing early
warning systems and potentially stopping future intrusions from escalating. The post-processing stage of the correlation engine also allows the generation of a historical database of alerts and signatures that can increase the effectiveness of the system.

### 14.6.3.1 Alert Prioritization

The purpose of alert prioritization is to classify alerts based on their severity and take appropriate actions for dealing with each alert class. Usually this operates as a means of finally assessing security incidents and ranking them in terms of known or expected damage.

In [52], alert prioritization is performed using two parameters: (a) The degree to which an alert is targeting a critical asset or resource, and (b) the amount of interest the user has registered for this class of security alert. Now the high priority incidents are identified for the environment within the organization within five grades from low priority to high priority. The final rank for any incident is the merging of the likelihood value and priority estimation.

### 14.6.3.2 Intention Recognition

Intention recognition is the process of inferring the goals of an intruder by observing their actions. This step aims to provide early warning capability and allow automatic response as well as preventing intrusions from escalating [53]. Offline data is inspected to allow a link between actions and intrusions to be determined. Intention recognition has also been considered for unique attacks in WLANs [54].

There are some issues that need to be overcome before proper intention recognition can be implemented in a network security situation [55]. The first of these issues is the tendency for attackers to try to cover their tracks. A malicious source can aim to take evasive action or masquerade in order to avoid discovery. This makes identification and root causing problematic. Second, there are practical limitations on the information sources:

- Holes in IDS coverage/unobserved actions
- Partial ordering of attack approaches
- Multiple attacker goals/effects
- Multiple hypotheses for attack intent in any situation

The plan recognition system proposed in the paper creates hierarchical options for an attacker, who is assumed to have an attack plan and does not just launch arbitrary attacks. Recognizing the plans of a hostile adversary requires implications and deductions rather than binary certainty. Due to this fact, it is difficult to recognize attacks that have long time scales. In order to compensate for unobserved events, the system may generate possibilities based on the observed actions.

### 14.7 EVALUATION

The final stage of the WIDS hierarchy defined here is evaluation. At this stage, decisions need to be made either by a human or an automated response system about the severity, likelihood, and impact of the alerts that are generated. It is important for the system to be able to readily identify the presence of a specific attack occurring at a specific time and the reasons for this alert in order to best inform any response or audit mechanism. Each of the planning decisions from previous steps directly contributes to the success of the outcome of the WIDS at this point; the selection of correct threats, architecture choices, metric and data collection identification, attack detection algorithm development, and intelligent correlation methods are employed (Figure 14.14).

Evaluation in the context of a WIDS covers two topics:

- Evaluation of WIDS–generated alerts
- Evaluation of the performance of the WIDS
14.7.1 Evaluation of WIDS Alerts

The key components for administrators or automated response systems are the following:

- The volume of alerts
- The confidence in the validity of alerts
- The ability to interpret alerts in a meaningful way

It has been mentioned in [56] that these human factors of intrusion detection are more important in industry than the technology challenges, although they are interrelated. The goal at the evaluation stage for a WIDS is to reduce the amount of data displayed to the administrator, to display it in a meaningful way, and to make sure the administrator has confidence in the output presented. The success of a human administrator in evaluating the probability of an attack is based on the following information:

- False positives
- False negatives
- Visualization
- Clarity of response action

14.7.1.1 False Positives

A false positive occurs when the attack detection algorithm identifies traffic as suspicious and/or malicious but that later turns out to be legitimate. This eventuality is the greatest source of frustration for users and designers of WIDSs and reduction of false positives is a critical goal.

If an alert requires further investigation to ensure it is a true positive, then the resolution response time suffers, potentially allowing the attack to perpetuate before confirmation. Conversely, if the WIDS is deemed trustworthy and an immediate response is carried out, then the reaction could conceivably cause more harm than the false alert itself. Hence, reduction of false positives can be dependent on a trade off again between response speed and thoroughness.

Some sources [57] identify the presence and volume of false positives as a critical stumbling block of WIDSs. Intolerance of this level of WIDS false positives has led to the development of intrusion prevention systems (IPS) as an alternative. While there is much improvement required, it is important to remember that most security systems create false positives, but it is how they are dealt with that determines the success of the system.

14.7.1.2 False Negatives

A false negative occurs when an intrusion is not detected by the system or is detected by the system but flagged as legitimate. The problem of false negatives is another issue for WIDSs although, generally, reduction in false negatives can be achieved by lowering threshold limits or by reducing the precision of detection rules. Unfortunately, this tactic is likely to drastically increase the level of false positives.

14.7.1.3 Visualization

Due to the complex interactions between metrics or network components, the root causing and visualization of alerts can be problematic. For alerts and recommendations passed on to automated detection systems, this area is of little concern other than for potential human auditing. However,
visualization of attack behavior and consequences is critical for any human observer in order to be able to make reasonable judgments about appropriate response.

Visualization techniques can be used to illustrate and characterize trends, events of interest, and incidents. This reduces the possibility of improperly interpreting the output of the WIDS and carrying out a potentially damaging, incorrect remediation activity. Intelligent visualization techniques represented in a timely, succinct, and meaningful format have the capability to aid the identification of and mitigation against false positives and false negatives.

Some open-source initiatives that provide IDS visualization include the following:

- **Graphviz**: Allows flowcharts and connected graphs to be automatically generated from simple text files.
- **EtherApe**: Graphical network analyzer for UNIX that displays the direction and volume of network traffic between hosts.
- **Netgrok**: A java implementation of a network analyzer that visually organizes network layout and data.

The focus of intrusive event visualization is largely related to graphic representation of traffic [61], topologies [59], decisions [60], and relationships in network activity. Demonstrating the performance of the WIDS in terms of detection performance is a method of visualization that is often overlooked and can include the following:

- **Accuracy**—In terms of percentage of detection, percentage of failure, and number of false positives.
- **Precision**—Number of predicted intrusions that were intrusions.
- **Recall**—Percentage of real intrusions covered by the system. This is quite difficult to gauge.
- **ROC curves**—ROCs (receiver operating characteristic) are detection visualization graphs that demonstrate the performance of a WIDS based on the link between false positives and true positives.
- **Timely response**—Display of the latency of alerting to an intrusion occurrence and/or speed of automated or human response.
- **Cost**—Calculating the cost associated with fighting an intrusion vs. the cost of the intrusion actually happening [61].

### 14.7.1.4 Response

At the top of the hierarchy, decisions need to be made about the severity, likelihood, and impact of the alerts that are generated as well as the response. Once an administrator can trust the alerts generated by a WIDS and can visualize the effect this alert is having on the network, the next step is identifying an appropriate response.

For interventions by humans, the choice of response is typically dependent on experience, and so responses can vary from person to person. This occurs due to the lack of comprehensive, effective response tactics to remediate many network attacks. Generation of these response tactics is a difficult task as serious consideration has to be put toward ensuring the response is proportionate and cost effective and does not cause more problems than the attack, such as a self DoS [61].

Recognition of this challenge has encouraged the development of IRSs (intrusion response systems), which are dealt with in greater detail elsewhere in this book. An IRS automates the human response behavior at the top level of a WIDS. Hence, many of the same issues with human response remain but with a technology rather than a human interpretation solution. This approach has many benefits, such as increase in response time, direct attribution of remediation to input data, and transfer of security responsibilities.
Implementation of an automated response strategy requires explicit trust in the performance of a WIDS and the tuning of attack detection parameters. In some cases, only those events that are classified with a high probability are dealt with by automated systems with uncertain events escalated to a human invigilator. In effect, this approach reduces the burden on human interpretation without removing it entirely. The same challenge remains in how to assist a human in making difficult security decisions.

14.7.2 Evaluation of WIDS Performance

There is no simple or standardized method of verifying the performance of a WIDS against benchmarks or a method of comparing WIDSs to each other [62]. There are no open standards for testing or any public, comprehensive test suites available. Hence, assessing the performance of WIDSs proposed by academia or commercial enterprises is difficult. Scarfone and Mell [10] advocate that any evaluation should be based on the following factors:

- Configuration ability and ease
- Burden load detection system requires
- Dependence on positioning
- Processing power of detection machines

Although testing of research systems has been carried out, it was criticized [63] as suffering from only being a simulation with no real world tests and difficult tuning. Challenges that arise from trying to test a WIDS effectively include the following [10]:

- No standard, open methodology
- Need for system to be tested in real-world environments
- There are no testing suites available
- Lack of lab environment test resources
- No configuration equivalence between WIDSs

Some notable example of WIDS evaluation are demonstrated by NSS Labs* [7,9]. NSS Labs are a commercial organization that has produced reports analyzing the performance of various security products, one of which is WIDS. The most recent test on this area is from 2001 [64], however, with more modern investigations focusing on IPSs (intrusion preventing systems). Furthermore, many of the reports require subscriptions or payment for access. In [7], a confusion matrix is constructed to allow the relative coverage areas of WIDSs to be compared using attributes, such as prosecution, confirmation, identification, recognition, and detection. In [9], another confusion matrix is developed, which purports to be able to differentiate WIDSs based on their attack performance using target and attack type although this is only proposed and not proven.

Nonetheless, for academic research purposes, the statement in [14] that “Exhaustive quantitative performance evaluations of currents IDSs in real-world environments do not exist” unfortunately still holds true.

14.8 SUMMARY

This chapter has categorized the typical operation of a common WIDS into six sections: threat identification, architecture considerations, data collection, detection strategy, correlation method, and evaluation. These six categories are relevant for any IDS although the focus in the descriptions has concentrated on wireless IDSs.

Discussion of the major attributes of each of these categories has demonstrated that the choice of IDS characteristics can influence the performance of subsequent stages. The field of “detection strategy” is the area of greatest current output in research; however, each of the remaining areas is either directly affected by or directly influences this stage in the process. Consequently, more work is needed to ensure that the data and recommendations produced by one stage are appropriate and meaningful for subsequent stages and, crucially, have easily evaluable and root causing components.

Maintaining a credible link between an indication of an attack occurring and comprehensible evaluation for a human administrator or automated response system should be the primary objective throughout the entire process. Future research should take into account the interrelationship between the stages and not solely consider them in isolation. Poor choices in the design of lower stages in the WIDS process can impact on the outcome of the entire system, leading to cascading suboptimal performance.

AUTHOR’S BIOGRAPHY

Jonny Milliken is a post-doctoral researcher at the Queen’s University Belfast (QUB), Belfast, UK. He was awarded an MEng (first class) from QUB in 2009 with a specialization in WiFi intrusion detection systems, and he holds CAPM and LCGI qualifications. He graduated from QUB in December 2012 with a PhD investigating WiFi intrusion detection strategies for public and open access WLANs. Jonny’s research interests include WiFi and cyber security, WiFi malware, testbed development, disaster response methods, and national infrastructure security, and his current work examines applications of WiFi for emergency search and rescue scenarios. He is also a member of IEEE and the IET and is involved with the IAESTE and ERASMUS programs in Northern Ireland.

REFERENCES

5. Simmons, C. et al., AVOIDIT: A cyber attack taxonomy, University of Memphis, Tennessee, August 2009.


57. Gartner, Gartner information security hype cycle declares intrusion detection systems a market failure, 2003.