

# Analysis of Distributed Intrusion Detection Systems Using Bayesian Methods

Daniel J. Burroughs, Linda F. Wilson and George V. Cybenko  
Thayer School of Engineering  
Dartmouth College  
Hanover, NH 03755

## Abstract

*In computer and network security, standard approaches to intrusion detection and response attempt to detect and prevent individual attacks. However, it is not the attack but rather the attacker against which our networks must be defended. To do this, the information that is being provided by intrusion detection systems (IDS) must be gathered and then divided into its component parts such that the activity of individual attackers is made clear. Our approach to this involves the application of Bayesian methods to data being gathered from distributed IDS. With this we hope to improve the capabilities for early detection of distributed attacks against infrastructure and the detection of the preliminary phases of distributed denial of service attacks.*

## 1 Introduction

Attackers in cyberspace benefit greatly from the anonymity, speed, and vast amounts of information present in that environment. Moving from one computer to another, obfuscating the source of the attack, attackers are able to make themselves difficult to trace. In certain cases, it is even possible to falsify information that would normally provide a link back to the attacker. Automated tools make it possible to scan and attack vast numbers of hosts on the Internet, pausing only briefly at each one. When an attack does occur, it may be only a matter of seconds before a system is compromised. Given a little more time, the attacker is able to cover up the evidence of the intrusion. To add to this already complex system, intrusion detection systems must be able to detect and prevent attacks while allowing vast amounts of information to travel around the various networks at incredibly high speeds.

Most current approaches to intrusion detection attempt only to detect and prevent individual attacks. However, it is not the *attack* but rather the *attacker* against which networks must be defended. Through

understanding the behavior of the attacker, it is possible to develop a clearer picture of what is occurring. To do this, the information being provided by intrusion detection systems must be gathered and then divided into its component parts such that the activity of individual attackers is made clear.

In this paper we discuss the current state of our research into using Bayesian multiple hypothesis tracking [2,12] as a basis for identifying the activities of individual attackers as they move across many networks. The main goal of this work is to improve the understanding of the attackers' behavior by using the existing data that is already being collected by intrusion detection systems scattered across several networks. By treating the intrusion detection systems as a sensor web, and applying mature concepts from sensor fusion techniques and target tracking algorithms, we aim to generate a higher level of situational awareness. This can then aid in the process of defending against attackers by providing insight into their motives and methods.

## 2 Background

### 2.1 Intrusion Detection Systems

Intrusion detection systems (IDS) are defined by both the method used to detect attacks and the placement of the IDS on the network. An IDS may perform either *misuse detection* or *anomaly detection* and may be deployed as either a *network-based* system or a *host-based* system. This results in four general groups: misuse-host, misuse-network, anomaly-host and anomaly-network. Some IDS combine qualities from these categories (usually implementing both misuse and anomaly detection) and are known as *hybrid* systems.

Network-based intrusion detection involves a network device listening to all the traffic that is going through its local neighborhood. Placement of the sensor is important since devices that direct traffic, such as switches and routers, separate the network into seg-

ments. A sensor on a segment sees only the traffic that is either going to or coming from other hosts on that segment, thus giving each sensor a limited view of its environment.

Host-based IDS examine the activity on a specific host. This allows them the advantage of having greater access to the logs and files of a particular computer, while being limited in what external activity they can see [9]. This limits the breadth of the sensor’s view, yet allows it to see greater depth and detail.

Misuse detection models attempt to match activity occurring on a network or host to predefined patterns or signatures. They compare current activity on the system against known patterns of attack in a process similar to the behavior of a virus scanner [8]. The strength of a misuse detection model is that it has a relatively low rate of false positives. However, it is limited to detecting only those attacks for which it has a signature, leaving it vulnerable to new attacks.

Anomaly detection models operate by building a model of system behavior based upon the standard operation of the network or component under observation. After this model of “normal” system behavior has been created, current activity is compared to it. When the deviation grows greater than a threshold level, an alert is triggered [9]. Such a system has the advantage of being able to detect attacks that are not currently known. The drawback of such systems is that they often have a high false positive rate, which can lead to a lack of trust in the software. These systems may also be defeated by malicious activity that masquerades as acceptable behavior.

## 2.2 Related Work

There is a great deal of work that is currently being performed in the area of intrusion detection. Much of the work centers around improvement in the ability of systems to detect attacks and the speed of network traffic that can be handled.

There are several projects dedicated to the collection of IDS data. One of these is the SANS Institute’s Incidents.org project [10]. This is a central collection site for intrusion detection data that is being collected from numerous volunteer sites. A number of groups are also working towards greater interaction between intrusion detection systems and improved analysis of collected data. Among these are the EMERALD project at SRI [11], work by Dain and Cunningham at Lincoln Labs [3], the Common Intrusion Detection Framework (CIDF) [6], and the Internet Engineering Task Force’s Intrusion Detection Exchange Format (IDEX) [4, 5].

## 3 Data Refinement and Knowledge Creation

The goal of this work is not to improve upon the methods currently used in intrusion detection, but rather to develop methods for more effectively analyzing the information already being provided by existing intrusion detection systems. Our goal is to be able to reorganize the existing data such that related incidents become apparent.

### 3.1 Network vs. Attacker Centric View

When the primary concern is the defense of a host or network, it is natural to adopt a view placing the object to be defended at the center. This is the view that best describes the implementation of most intrusion detection systems. Networks are built with the concept of a perimeter, consisting of firewalls, border routers, and gateways. This creates a wall around the network, limiting external access to strictly monitored channels. Since this limits the entry points which an attacker may use to gain access to a network, defensive systems are usually located at these boundary points. It is possible, as is often the case, to defend a network or portion of a network in isolation from any other network. While this is a sensible approach for defense, it does little to aid in understanding the methods and motives of the attackers in general.

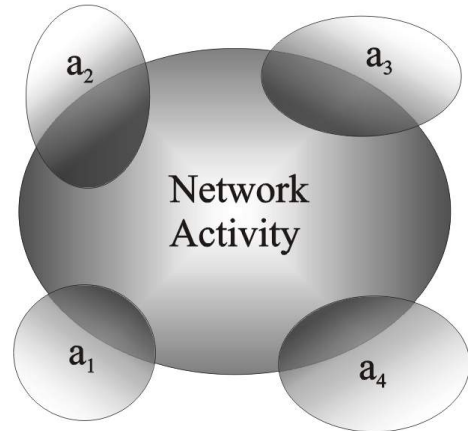


Figure 1: Network Centered Viewpoint

Figure 1 shows activity graphs for a network (N) and four attackers ( $a_1, a_2, a_3, a_4$ ) who are currently attacking the network. When we limit our view, and therefore our data collection capabilities, to this network, we are able to see only the portion of each attacker’s activity which intersects the activity observable on this network. This is further complicated by the lack of complete visibility of our network and by false alarms caused by the sensors.

To gain a more complete picture of an attacker’s action, we must expand our view to include more than one network and more than one type of defense system. Rather than center our viewpoint around a single network, we gather and analyze data from many distributed systems in order to obtain a more complete picture of the attackers’ activities.

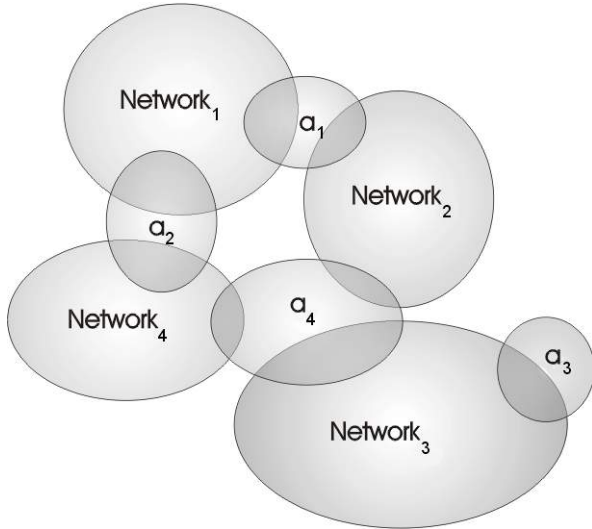


Figure 2: Attacker Centered Viewpoint

A single attacker’s actions may coincide with several networks as displayed in Figure 2. By gathering information from multiple networks, we are able to expand our view of an individual attacker’s actions. However, simply gathering this information from multiple networks does not provide much useful information. The sensor reports must be reorganized such that the activities of individual attackers are made clear. Described below are the methods used to do this.

### 3.2 Observe, Orient, Decide and Act

In information warfare (IW), the decision-making process is governed by the observe, orient, decide and act (OODA) loop developed by Boyd [13]. The second stage of this loop, the orient stage, is a process of knowledge creation. In this process, sets of similar or dissimilar data are aligned, correlated and combined to model, explain, and predict the behavior of the system. The Department of Defense Joint Directors of Laboratories have broken the process of data fusion into five levels of refinement: data, object, situation, meaning, and process [13].

In the data refinement stage, raw data is gathered and preliminary analysis of the data is performed. Noise is removed, data is limited to areas of interest,

and initial object detection is performed. In this stage we are using intrusion detection systems to gather raw information regarding the attacks occurring on our systems. Initial filtering is done in order to reduce the amount of information being gathered and to group related events. For example, port scans of the network or operating system fingerprinting attempts that would normally have triggered IDS rules many times over could be reduced to single events.

In the object refinement stage, data is normalized and described in a common format. This includes time synchronization and conversion to a common description format such as the DARPA Common Intrusion Detection Framework (CIDF)[6, 7] or the Internet Engineering Task Force’s Intrusion Detection Exchange Protocol (IDXP)[5, 4]. In these formats, intrusion events are described as an object that has a set of attributes. We are concerned with features such as the time of the event, internet protocol (IP) source and destination addresses, service under attack, type of event (i.e., port scan, buffer overflow), etc. Once these steps have been completed, tracking and identification of the targets are performed.

During the situation refinement stage, aggregate sets of the objects are detected by common or related behavior. Some correlations are very simple, relying on common attribute values across a number of events. Examples of this would include a multitude of attacks all originating from the same IP address, or a number of similar style attacks coming from varied IP addresses within a short period of time. However, other relationships will not be as straightforward and will require models or patterns describing attacker behavior, which are compared to the events that are being detected by the IDS. By doing this, we are moving away from determining what individual events are being seen and are heading toward an understanding of what is happening in the bigger picture. The methods used to correlate the data are described in Section 4.

During the meaning refinement process, situation knowledge is used to model and analyze possible future behaviors of the objects and groups built in the previous stages. Finally, current knowledge of the situation is compared to the knowledge required to achieve one’s goals. This is done to determine shortfalls in the knowledge base and then minimize them through process refinement.

## 4 Multiple Hypothesis Tracking

The situation refinement stage, as described in Section 3.2, is accomplished through the use of a Bayesian multiple hypothesis tracking (BMHT) algorithm [13]. BMHT is a method of target tracking that allows de-

cisions to be adjusted and refined until enough data has been collected to ensure a level of confidence. In its basic form, the algorithm generates and stores all possible hypotheses that could explain the data being measured. To determine the likelihood that a particular hypothesis is correct, it is evaluated against our understanding of the sensor behavior and the dynamics of the target. After all hypotheses have been evaluated, the one with the greatest likelihood is assumed to be correct [1]. As new data arrives, the likelihood of each hypothesis is adjusted and our belief in that hypothesis is either strengthened or weakened. This makes BMHT particularly useful when it is necessary to perform real-time target tracking with incomplete or inaccurate data.

#### 4.1 Hypothesis Generation

Each hypothesis consists of a set of tracks that map events measured by sensors to targets. A track is a series of events that describes the motion, or activities, of an individual target. When a new event occurs, it may be assigned to an existing track, create a new track, or be considered a false alarm and not assigned to any target track. Within a hypothesis, each event appears in exactly one track. This prevents a single sensor event from being assigned to more than one target. Each hypothesis has a likelihood value that is based on its set of tracks, the dynamics of the target and the performance of the sensors. The number of possible hypotheses grows extremely large as more events are measured. For example, if we have two events, a port scan for DNS servers and a buffer overflow attack against DNS servers, there are five possible hypotheses as shown in Table 1.

Table 1: Set of Possible Hypotheses for Two Events

	Scan	Attack
H-1	False Alarm	False Alarm
H-2	Target 1	False Alarm
H-3	False Alarm	Target 1
H-4	Target 1	Target 2
H-5	Target 1	Target 1

The large number of potential hypotheses makes it necessary to impose limits on what is kept around for future evaluation. One method of doing this is to introduce a threshold such that when a hypothesis' likelihood falls below the limit, it is deleted. Likewise, individual tracks may be removed as well. Track removal takes on two forms: track deletion and track completion. Track deletion is similar to hypothesis deletion; when a track is considered too unlikely to

be true, it is removed. Track completion occurs when a track is likely to be correct, but is not expected to have any more events added to it. It is important not to assign a track as being completed too soon, as it will prevent future events from being included in that track. However, this type of error is recoverable by later comparing sets of completed tracks. If an attacker's actions are broken up into multiple, completed tracks, it is possible to later recombine these tracks into one large track. It is important that we use techniques such as these to reduce the volume of information required by the BMHT algorithm.

#### 4.2 Likelihood Evaluation

Let  $y$  be the set of sensor readings defining a track and let  $x$  be the set of target states we believe to have caused these readings. Two determinations must be made: the likelihood of having received readings  $y$  given state  $x$ , and the probability of the target existing in state  $x$ . Through the use of Bayes' theorem [12], this gives us

$$p(x|y) \propto L(y|x)p(x). \quad (1)$$

If multiple observations are made of the target, and each sensor has an independent likelihood function ( $L_1, L_2, \dots, L_n$ ), the overall probability can be calculated as

$$\begin{aligned} p(x|y_1, y_2) &\propto L_2(y_2|x)L_1(y_1|x)p(x) \\ &\propto L_2(y_2|x)p(x|y_1). \end{aligned} \quad (2)$$

This process may be repeated any number of times. Thus, in the case where all observations are independent, Bayes' Theorem is naturally recursive, allowing us to compute the new posterior distribution from the previous posterior distribution. A full explanation of the BMHT algorithm is beyond the scope of this paper, but it is well presented in [12].

#### 4.3 Attacker Behavioral Models

To evaluate a track, both the performance of the sensor and the behavior of the target are considered. These may be performed independently and are known as the sensor update and the motion update. The sensor update,  $L(y|x)$ , is often well understood for misuse detection IDS. However, the motion update evaluation,  $p(x)$ , requires a probabilistic evaluation of the target's motion. Different attackers will have different goals, and thus their motion through the attack state space will follow different paths. If an attacker is trying to be stealthy and avoid detection, the methods of attack chosen will differ from an attacker who is going for speed and number of compromised machines.

The attacker behavioral models describe the series of actions an attacker is likely to use while attempting to reach his or her goal(s). The feature set used to describe an attacker’s motion include the particular technique or vulnerability being used (e.g., BIND <sup>1</sup> buffer overflow, IIS Unicode attack <sup>2</sup>, etc.), source and destination IP addresses, destination port (which defines the service being attacked), time of the event, and so forth. The attackers’ behavior may then be described as a likelihood of moving through these attributes. This may be measured by instantaneous value (what is happening at this event only), as a sequence of events, or as a rate of change.

An attacker trying to gain zombie computers for a denial of service (DoS) attack will want to move quickly through as many computers as possible. If we look at the destination IP address for IDS events triggered by this activity, a high rate of change (RoC) would be evident. However, since the attacker is more interested in speed and number of compromised systems, usually only a single type of attack is used. This leads to a very low RoC in attack technique. If instead an attacker were attempting to gain control of a specific machine, we would expect to see the attacks centered around that machine. The attacker may attempt to gain control of other nearby machines in order to exploit a trusted relationship between the two, so the attack need not be limited to just the target. Also, it is likely that the attacker will make multiple attempts to break into the target host, using multiple methods of attacks, and target various services on that machine. These two examples contrast with the events we would expect to see during a denial of service attack. In this case, the source IP is often spoofed to make it appear as if the attack is coming from many different places. While the events occur at a very rapid rate, the types of events tend not to change. Table 2 details the characteristics exhibited by DoS, zombie collection, and directed attacks.

From this, we develop probability distributions representing the motion characteristics of the attackers. These may then be used to evaluate the likelihood of a event belonging to a particular track, and thus also evaluate the overall likelihood of the hypothesis. In order to maintain performance and develop generalizable models, we attempt to limit the number of attack features used in the modeling. Source address, destination address, service under attack, type of attack, and time of the attack are used in developing

---

<sup>1</sup> Berkeley Internet Name Domain (a DNS implementation).

<sup>2</sup> Microsoft’s Internet Information Server.

the models. Using these features, we attempt to distinguish between various types of attacks and various attackers.

## 5 Status of Work

Our initial testing environment is a network with an address space of 1000 hosts. It has significant numbers of hosts that are regularly used as well as areas of unassigned IP addresses. The network is divided into five sections, which we are using to represent five separate networks. We are analyzing IDS data coming from network IDS (Snort and SHADOW) distributed across these five subnets. In addition to the normal machines that are on the network, we have a number of *honeypot* systems. These are vulnerable machines that serve no purpose other than to be attacked. This is done to make a target inviting to potential attackers, so that they may aid in our research by providing us real-world data. In later stages multiple, distributed networks will be used for data collection and performance evaluation.

In preliminary tests, we used the testing network to gather real-world background noise data while generating simulated attacks against machines on that network. While not as accurate as completely real-world data, this allowed us to be able to better control the data being analyzed. This is particularly useful during the developmental stages.

The system was tested using 5 simultaneous attack scenarios containing roughly 800 events. In addition, 200 non-scenario events were included as background noise. The tracking system was able to correctly place 89% of the scenario events into the correct scenarios. However, 20% of the non-scenario events were incorrectly included in the scenarios. This is the amount of information that was misclassified as being part of an attacker track while in fact it was not. It is important to note that false tracks were not created, but rather the information that was included in each track was not entirely correct.

We believe that the tracking system could be improved through the use of a multiple pass technique. In the initial passes, easily identifiable groups of events would be gathered together to reduce the overall data volume. Instead of scanning attempts being seen as many individual events, they would be replaced by a single, more descriptive scan event. This would reduce the computational and memory requirements of the tracking problem. Thus, instead of getting bogged down with the large amount of easily classifiable events, the tracking system could concentrate on the events which are harder to classify.

Table 2: Modeling Types of Attacker Behavior

	Denial of Service	Zombie Collection	Directed Attack
Technique RoC	Low	Low/None	High
Source IP RoC	High	Low/None	Low
Dest. IP RoC	Low/None	High	Low
Dest. Port RoC	Unknown	Low/None	Medium
Time Rate of Events	Quick	Unknown	Unknown
Type of Events	DoS	Scan Remote-Access	Reconnaissance Scan Remote-Access Privilege Level

## 6 Future Work and Conclusions

The next stages of development include expanded testing and evaluation, development of multi-pass analysis techniques, integration with other types of intrusion detection systems and automated techniques for attack pattern generation. A multi-pass system would allow for quick local correlation, followed by more extensive global correlation. By integrating other types of intrusion detection systems, such as host-based IDS, the tracking system will be able to take advantage of different views of the same targets. By fusing data from various types of IDS that are all observing the same domain, the tracking system will have a more complete view of the behavior and actions of the attacker. Finally, the generation of attack patterns will be addressed. Currently, these are created manually. It is desirable to use machine learning techniques to build these from historical data sets.

We have presented a method for using Bayesian multiple hypothesis tracking to classify intrusion detection system events into attack sequences. This may be used to reorganize data that is already being collected from intrusion detection systems in order to provide security analysts with a better *situational* view of what is occurring on their networks. By doing so, the actions of individual attackers are made clear so that the proper steps to minimize the potential damage and losses due to attack may be taken as rapidly as possible.

## References

- [1] C. Alberola and G. V. Cybenko, "Tracking with text-based messages", *IEEE Intelligent Systems*, pp. 70-78, 1999.
- [2] Z. Chair and P.K. Varshney "Distributed Bayesian hypothesis testing with distributed data fusion", *IEEE Transactions on Systems, Man, and Cybernetics* Vol. 18, No. 5:695-699, 1988.
- [3] O. M. Dain and R. K. Cunningham, "Fusing heterogeneous alert streams into scenarios", *Proceedings of the Eighth ACM Conference on Computer and Communications Security* Philadelphia, PA, 2001.
- [4] Internet Engineering Task Force, "Intrusion detection exchange format data model", <http://www.ietf.org/internet-drafts/draft-ietf-idwg-data-model-03.txt>, 1999.
- [5] Internet Engineering Task Force, "Intrusion detection exchange format requirements", <http://www.ietf.org/internet-drafts/draft-ietf-idwg-requirements-02.txt>, 1999.
- [6] C. Kahn, P. Porras, S. Staniford-Chen, and B. Tung, "A common intrusion detection framework", Submitted to the *Journal of Computer Security*, 2000.
- [7] W. Lee, R. Nimbalkar, K. Yee, S. Patil, P. Desai, T. Tran, and S. Stolfo, "A data mining and CIDF based approach for detecting novel and distributed intrusions", *RAID 2000*, 2000.
- [8] M. J. Ranum, "Intrusion detection: Challenges and myths", [http://secing.net/info/ides/index\\_mythe.html](http://secing.net/info/ides/index_mythe.html), 2000.
- [9] SANS Institute, "Intrusion detection FAQ", [http://www.sans.org/newlook/resources/IDFAQ/ID\\_FAQ.htm](http://www.sans.org/newlook/resources/IDFAQ/ID_FAQ.htm), 2000.
- [10] SANS Institute, "Incidents.org", <http://www.incidents.org>, 2001.
- [11] SRI, "EMERALD intrusion detection system home page", <http://www.sdl.sri.com/projects/emerald>, 2001.
- [12] L. D. Stone, C. A. Barlow, and T. L. Corwin, *Bayesian Multiple Target Tracking*, Artech House, Norwood, MA, 1999.
- [13] E. Waltz, *Information Warfare: Principles and Operations*, Artech House, Norwood, MA, 1998.