# Fusion and Filtering in Distributed Intrusion Detection Systems

Paul Barford Somesh Jha Vinod Yegneswaran
pb@cs.wisc.edu jha@cs.wisc.edu vinod@cs.wisc.edu
University of Wisconsin, Madison

**Abstract**

False alarms and timely identification of new attacks are two of the biggest challenges to the effective use of network intrusion detection systems (NIDS). A potential means for addressing these shortcomings in modern NIDS is employing multiple, distributed network intrusion detection systems (DNIDS). In this paper we consider the potential benefits of DNIDS by addressing two open problems. The first problem is how to combine data from multiple intrusion sensors in a network. This is known as the fusion problem. The second problem is how to identify the most important data provided by multiple sensors in a network. This is known as the filtering problem. We develop a series of analytic and simulation models to assess the potential benefits of DNIDS for reducing false alarms and improving timeliness of detection for different fusion and filtering strategies. Our analysis explores the trade-offs when fusion and filtering are used together and shows that significant improvements are possible.

## 1 Introduction

Malicious traffic in the form of self-propagating worms, viruses, port scans and denial of service attacks is a profound threat to both the Internet itself and to the infrastructures that use the Internet for communication. Unfortunately, protecting networks from malicious traffic remains a vexing problem in both the research and operational communities. Current best practices for protecting networks from malicious traffic are to deploy a multi-layered security infrastructure that includes network intrusion detection systems (NIDS). While NIDS are useful for identifying malicious activity in a network, they generally suffer from two major drawbacks: high false alarm rates and perspective from a single vantage point, which limits their ability to detect distributed or coordinated attacks.

One promising approach to addressing the above-mentioned shortcomings is through the use of *distributed network intrusion detection systems* (DNIDS). A DNIDS is composed of diverse set of sensors that coordinate to identify malicious traffic. Sensor diversity appears in three forms: *information* , *temporal* , and *geographical*. Different sensors can measure different features of network traffic or might use different detection algorithms *i.e.,* they can provide fundamentally different information about the network traffic (hence the term information diversity), which can be used to improve intrusion detection capabilities. Since many attacks consist of multiple steps, sensor measurements can become correlated across time (hence the term temporal diversity). Correlating sensor readings or events that occur at different times can also lead to improved intrusion detection. Several classes of malicious traffic (such as worms) can have a widespread effect in the network. Thus, sensors that monitor different sets of network addresses will have a "similar" affect on their measurement under such widespread attacks (hence the term geographical diversity). In the context of intrusion detection, geographical diversity can be used to improve both the time-to-detection and false-alarm rate. Our focus for this work is to investigate how to combine data from geographically diverse DNIDS in order to improve false-alarm rates and timeliness in detecting attacks.

In this paper, we use simple models for summarizing (*i.e., filtering*) and combining (*i.e., fusing*) data in DNIDS in order to explore the problem space. Our approach is to consider statistical (as opposed to signature-based) intrusion detection using thresholding, *i.e.,* an alarm is raised if a sensor reading goes above a specified threshold. In such a system, a false alarm is defined as an alarm raised during normal conditions. The primary reason for false alarms in statistical detection systems is the variance of network traffic during normal operations. Therefore, *to reduce the false-alarm rate of statistical detection in DNIDS it is very important to reduce the variance of the normal traffic*.

We assume that we have $N$ identical sensors located at different geographical sites then an alarm is raised at a central site if the average reading of the sensors exceeds a threshold $t$. Since the sensors are deployed at different sites, averaging their readings should reduce the variance of the normal component of the traffic. Moreover, since the attack component of the traffic typically is correlated at different sites, averaging should not affect (sometimes increases) the variance of the attack component. Thus, the questions become what is the granularity of the data that should be passed to the central site and how should the data be combined at the central site? To that end, this paper makes the following contributions:

- **Analytical formulation:** Section 3 provides an analytical formulation for studying the reduction in the false-alarm rate by averaging readings across various sensors.
- **Simulation analysis:** Section 4 provides a simulation-based evaluation of the impact on false alarm rate and timeliness of detection using two different methods for filtering and fusion.

# 2 Related Work

## 2.1 Network Traffic Characteristics

General properties of Internet traffic have been studied intensely for many years - standard references include [8, 22, 14]. These studies have been focused on establishing baselines or invariant properties for typical network traffic behavior in the wide area. We draw on these studies both as a means for understanding the characteristics of benign network traffic and to inform our choices for generating traffic in our simulation experiments. One of the most important results relating to our work is that typical network traffic has self-similar scaling properties (*i.e.,* high variability) which significantly complicates detection of anomalies via statistical means.

A good deal of work has also been done to characterize Internet intrusion and attack activity. Moore *et al.* [19] examined the prevalence of denial-of-service attacks using backscatter analysis. In [17], the authors analyze the details of the Code Red worm outbreak and provide important perspective on the speed of worm propagation. In a follow-on work, Moore *et al.* [18] provide insights on the speed at which counter measures would have to be installed to inhibit the spread of worms like Code Red. In [35], Yegneswaran *et al.* explore the statistical characteristics of Internet intrusion activity from a global perspective. A related study of unwanted Internet traffic, *i.e., Internet background radiation*, is presented in [21]. Background radiation consists primarily of scans and the remnants of worms traffic that continually cycle throughout the Internet. Finally, the profiles of worm outbreaks have been characterized in several papers including [27, 7, 16]. An important contribution of these studies is the development of models for the propagation and scanning behavior of worms. In particular, the authors in [27] demonstrate that the equations used to model epidemic growth rates in finite systems of equally vulnerable entities can be naturally adapted to the domain of Internet worms. These results also inform our simulation models.

## 2.2 Anomaly, Attack and Intrusion Detection

A wide variety of techniques for detecting anomalies, attacks and intrusions have been developed and investigated. The most common of these is misuse detection which is employed in standard network intrusion detection systems including Snort [24] and Bro [23]. Wang *et al.* describe a interesting method for detecting SYN flood attacks based on change point detection [32]. Statistical methods for network anomaly detection, the focus of our work, have been investigated in [15, 26, 33]. Other anomaly detection methods include information retrieval [2], data mining [13], inductive learning [28], statistical signal processing based on change point detection [29], and multiresolution analysis [5]. For a detailed treatment of network intrusion detection, the reader can consult surveys and textbooks including [1, 20]. While all of this literature is related to our study, our specific focus is on investigating the problem space related to combining and analyzing data in *distributed* network detection systems.

The problem of event recognition in distributed sensor environments continues to receive attention in both the signal processing and sensor networking communities. Our intention to to investigate the utility of the techniques developed in these studies in the specific context of Internet intrusions and attacks. For example, Barkat and Varshney treat the problem of identifying events in a distributed radar system where each device makes a local detection decision and sends these results to a fusion center which then merges this data to make a final detection decision using a $k$ out of $n$ voting rule [6]. Chamberland and Veeravalli treat a similar problem in context of sensor networks in [9]. A study with a similar objective as ours but a different approach is described in [4]. That work proposes a change point detection heuristic for identifying attacks using data from multiple sensors.

The general aspects of distributed network intrusion detection systems (DNIDS) have been treated in a number of papers including [25, 3, 11, 12]. These studies typically treat the problems associated with physical infrastructure required for measurement and communication in DNIDS as well as some of aspects of treat detection in DNIDS. EMERALD is an example of a modular DNIDS that includes both monitoring and analysis components [31]. The analysis component of EMERALD uses both signature-based and Bayesian methods to detect intrusions. Dshield.org [30] takes a completely different approach to the DNIDS problem. Where as the systems noted above (for the most part) target a centralized security management authority, Dshield accepts logs from NIDS and firewalls around the Internet, aggregates them and then reports summaries on intrusion and attack activity. System administrators can use this information to configure and tune their security infrastructure. Finally, commercial entities such as Symantec offer managed security and early warning services to customers based on the use of DNIDS [10]. While each of these systems and studies offers a perspective on the problem of distributed intrusion detection, none has taken a first principles approach to evaluating the capabilities or utility of these systems against specific classes of threats.

## 2.3 The DOMINO Infrastructure

In prior work we have developed a novel distributed intrusion detection infrastructure which we call DOMINO (Distributed Overlay for Monitoring InterNet Outbreaks) [34]. DOMINO is a secure, scalable, robust data sharing infrastructure composed of a heterogeneous set of intrusion sensors distributed around the Internet. DOMINO's objective is to improve timeliness and accuracy of detecting malicious traffic. The infrastructure itself is composed of a hierarchical set of systems with decreasing levels of trust. As shown in Figure 1, DOMINO participants include:

- **Axis overlay:** The central component of DOMINO responsible for the bulk of data sharing and connected via a peer-to-peer overlay network. All communication between the axis nodes and between axis nodes and the satellites is authenticated, encrypted and facilitated via the DOMINO summary exchange protocol.

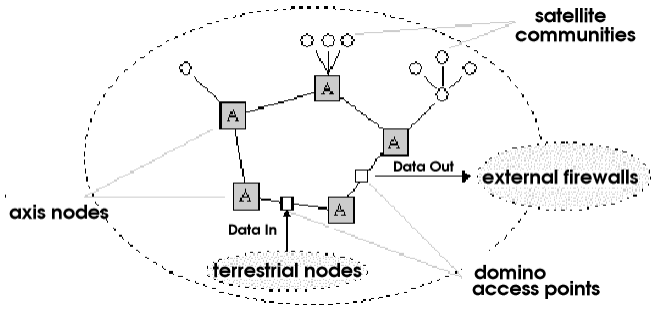


Figure 1: DOMINO Node Organization. Axis nodes participate in a peer-to-peer overlay network while satellite and terrestrial nodes form a hierarchy below.

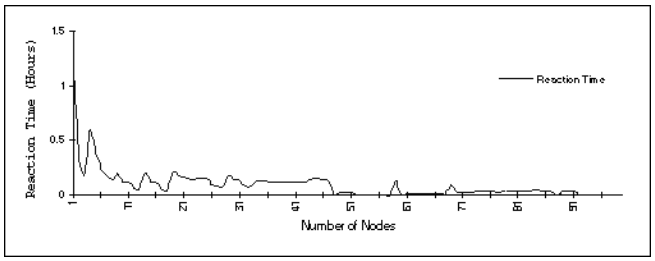


Figure 2: Empirical evaluation of reaction time for the SQL-Snake outbreak as more networks are added to a distributed detection system.

- **Satellite communities:** Smaller networks of nodes that implement a version of the DOMINO communication protocol. These nodes are organized as a hierarchy below an axis node or another satellite node. Data from these nodes is considered to be less trustworthy than from axis nodes.
- **Terrestrial contributers:** The least trustworthy nodes in the network, but potentially a large source of data.

The importance of DOMINO to this work is that it provides an environment to empirically test and evaluate the fusion and filtering functions.

We performed an off-line analysis of the potential for a distributed intrusion detection system like DOMINO for reducing the *reaction time* for attack detection in [35]. That study used a set of logs collected by Dshield.org during the time of the SQL-Snake outbreak in May 2002. (That worm was not a rapid propagator compared to prior outbreaks like Code Red and Nimda.) To perform our evaluation, we randomly selected 100 /24 networks and calculated the hourly average number of scans and the average number of sources using port summary data of port 1433 (MS-SQL) for the first two weeks of May. We define *reaction time* as the elapsed time between the *outbreak inflection point* and the first alarm after that point. The simple rules we chose for alarm generation were:

- 200% increase in number scans from hourly average, and
- 100% increase in the number of sources from hourly average, and
- number of sources is greater than five.

Figure 2 shows the **decrease in observed reaction time** from an average of more than an hour with a single node to almost zero as we add sufficient axis nodes (approximately $50$).

## 3 Analytical Model

Let $X^N$ and $X^A$ be random variables that model some feature of the network traffic during normal and attack conditions. Assume that the random variables are non-negative (which is true for most measurements, such as number of scans at a certain port). The specific feature modeled by the random variables depends on the attack being considered. For example, for detecting worms the random variables might represent the number of scans at the port that the worm exploits.

For the rest of the section, we assume that we have a sensor that monitors the specific feature of the network traffic being modeled by $X^N$ and $X^A$. Assume that the sensor raises an alarm using the following simple thresholding scheme: *Raise an alarm if the sensor measurement is above a threshold*

$t$. In this case, the false-alarm ($FA$) and the false-negative ($FN$) rate are given by the following expressions:

$$\begin{aligned} FA &= P(X^N > t) \\ FN &= P(X^A \leq t) \end{aligned}$$

False alarm rate is defined as the probability that an alarm *is raised* during normal conditions. Analogously, the false-negative rate is defined as the probability that an alarm *is not raised* during conditions that a network is under attack. For intrusion detection, the objective is a low false-alarm and false-negative rate.

Let $X$ be a random variable. Chebyshev's inequality is shown below:

$$P(|X| \geq k) \leq \frac{E(X^2)}{k^2}$$

Another formulation of the Chebyshev's inequality is the following (where $\mu = E(X)$):

$$P(|X - \mu| \geq k) \leq \frac{Var(X)}{k^2}$$

Assume that the threshold $t$ is greater than $\mu = E(X^N)$. Using Chebyshev's inequality we get the following upper bound on the false-alarm rate:

$$FA \leq \frac{Var(X^N)}{(t-\mu)^2}$$

It is important to notice that the false-alarm rate is directly proportional to the variance of the random variable $X^N$. If we set the threshold $t$ equal to $\mu + r\sqrt{Var(X)}$, then the false-alarm rate is bounded by $\frac{1}{r^2}$. As $Var(X^N) \to 0$, $FA \to 0$, *i.e., as the variance of the normal traffic goes to* $0$*, the false-alarm rate goes to* $0$.

Assume that the random variable $X^A$, which models the specific network feature under attack conditions, is the sum of two random variables $X^N$ (measurement under normal conditions) and $Z$ (additional measurement caused due to an attack), *i.e.*, $X^A = X^N + Z$. In this case, we have the following sequence of equations bounding the false-negative rate:

$$\begin{aligned} FN &= P(X^N + Z \leq t) \\ &\leq P(X^N \leq t) P(Z \leq t) \\ &= (1 - FA)\ P(Z \leq t) \end{aligned}$$

The formula given above assumes that $X^N$ and $Z$ are independent. Notice that $P(Z \leq t)$ is the ideal false-negative rate (denoted by $FN_{ideal}$); we cannot hope to do better than observing the "pure" attack traffic without normal traffic added to it. Also, notice as $FA \to 0$, $FN \to FN_{ideal}$. Moreover, as $Var(X^N) \to 0$, $FA \to 0$. In other words, *as the variance of the normal traffic* $X^N$ *tends to* $0$*, the false-negative rate tends to the ideal false-negative rate.* Hence, reducing the variance of the normal traffic lowers both the false-alarm and false-negative rate.

## 3.1 Averaging sensor readings

Assume that we have $n$ sensors. Intuitively, these sensors measure network traffic for a disjoint set of addresses. Let $X_i^N$ and $X_i^N + Z_i$ be the reading of sensor $i$ under normal and attack conditions. We make the following assumptions:

- The normal measurement $X_1^N, \cdots, X_n^N$ are independent and identically distributed (iid).
- For all $i$ and $j$, $X_i^N$ and $Z_j$ are independent.

If $X_1, \cdots, X_n$ are random variables with finite variances $\sigma_1^2, \cdots, \sigma_n^2$, and $S_n = X_1 + \cdots, X_n$, then

$$Var(S_n) = \sum_{k=1}^{n} \sigma_k^2 + 2 \sum_{j>k} Cov(X_j, X_k) \tag{1}$$

In particular, if the $X_j$ are mutually independent,

$$Var(S_n) = \sigma_1^2 + \cdots + \sigma_n^2$$

During the normal conditions the average of the sensor readings is given by the following expression:

$$\overline{X^N} \;=\; \frac{X_1^N + \cdots + X_n^N}{n}$$

Using equation 1 the variance of $\overline{X^N}$ is $\frac{\sigma_N^2}{n}$, where $\sigma_N^2$ is the variance of the normal traffic $X_i^N$. Therefore, the false-alarm rate (which is given by the expression $P(\overline{X^N} > t)$) is bounded by $\frac{\sigma_N^2}{n(t-\mu)^2}$. Recall that the false-alarm rate for the single sensor rate was bounded by $\frac{\sigma_N^2}{(t-\mu)^2}$. Hence, *by averaging the false alarm rate was reduced by a factor of* $n$. As noted before, reducing the variance of the normal traffic reduces the false-alarm and false-negative rate.

During the attack conditions the average of the $n$ sensor measurements (denoted by $\overline{X^A}$) is given by the following expression:

$$\overline{X^A} \;=\; \overline{X^N} + \overline{Z}$$

In the expression given above, $\overline{Z}$ is given by the following expression:

$$\overline{Z} \;=\; \frac{Z_1 + \cdots + Z_n}{n}$$

Assume that $Z_i$s are identically distributed and have variance $\sigma_Z$. Let $\rho_{i,j}$ be the correlation coefficient between $Z_i$ and $Z_j$. Recall that $Cov(Z_i, Z_j)$ and $\rho_{i,j}$ are related by the following equation:

$$\rho_{i,j}\sigma_Z^2 \;=\; Cov(Z_i, Z_j)$$

Using equation 1 the variance of $\overline{Z}$ is given by the following expression:

$$Var(\overline{Z}) \;=\; \frac{1}{n^2}(n\sigma_Z^2 + 2\sigma_Z^2 \sum_{j>k} \rho_{j,k})$$

Assume that the attack component $Z_i$ are correlated, *i.e.*, for all $j$ and $k$, $\rho_{j,k} \geq \alpha$ for some $0 < \alpha < 1$. In this case, we have the following lower bound on $Var(\overline{Z})$:

$$Var(\overline{Z}) \;\geq\; \sigma_Z^2 \alpha$$

Notice that as $n \to \infty$, $Var(\overline{X^N}) \to 0$ and $Var(\overline{Z}) \to \alpha\sigma_Z^2$. In other words, *as readings from more sensors are averaged, the variance of the normal traffic goes down to* $0$ *and the variance of the attack traffic stays above a positive lower bound*.

## 4 Experimental Evaluation

We evaluate the performance trade-off of fusion and filtering strategies for worm detection in terms of accuracy and overhead. Our metrics for accuracy are 1) mean time to response (reaction time) and 2) average number of alerts generated during normal operations (false alarms). We estimate bandwidth overhead as a function of the number of data units exchanged between the participating nodes.

### 4.1 Traffic Model

We begin our simulation analysis by selecting models for both normal and worm traffic. Here $Z(t)$ refers to the additional traffic introduced due to the outbreak, $N(t)$ is the background noise traffic on a particular port and $X(t)$ is the aggregate traffic.

$$X(t) = Z(t) + N(t) \tag{2}$$

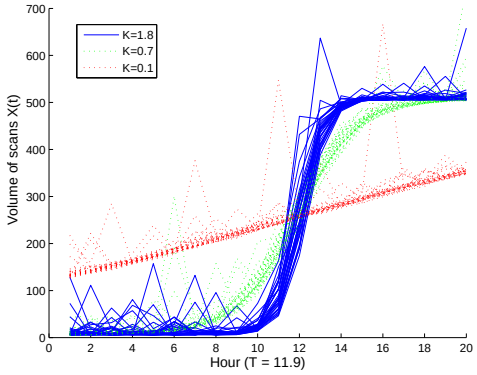


Figure 3: Simulated Background and Worm Traffic for different infection rates ($K$)

**Simulating worm traffic:** Two well studied models for simulating worm traffic include the simple homogeneous infection models [27] and the two-factor worm model which extends the Kermack-Mckendrick "classic epidemic model". The primary difference is that the latter also considers the clean-up of infected hosts [7]. We pick the former model for its simplicity and because we are primarily concerned with decisions during the growth-phase of worm outbreaks[1]. This growth rate of the volume of scans is expressed by the logistic equation described below.

$$a = \frac{e^{K(t-T)}}{1 + e^{K(t-T)}} \tag{3}$$

We assume that due to variability in network connectivity and topology, there are "lags" that affect when the inflection point of an outbreak is observed at certain networks. We assume that this offset follows a normal distribution. Thus the model for traffic during an outbreak becomes:

$$Z(t) = M \frac{e^{K(t+\epsilon-T)}}{1 + e^{K(t+\epsilon-T)}} \tag{4}$$

In the above equation, $K$ is the infection rate, $M$ is volume of scans observed during peak infection, $T$ is the inflection point. As a point of reference, the values for CodeRed outbreak where $T = 11.9$, $K = 1.7$ infections/hour. In our experiments, we fix $T$ and $K$ to these values except in cases where we explicitly state otherwise. The mean and variance of the offset function $\epsilon$ were fixed to be 0 and 0.3 respectively.

[1]The two-phase model might be more appropriate for very slow spreading worms like SQL-Snake where the percentage of susceptible population and correspondingly infection rate($K$) are very small.

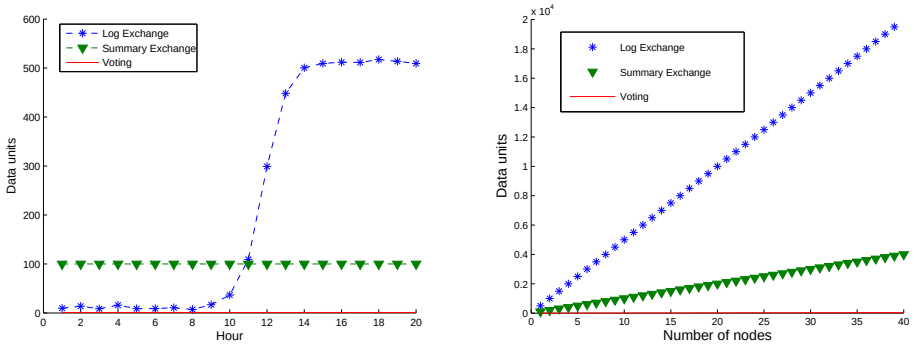


Figure 4: (a) Time-volume graph of hourly data exchanged with 20 nodes (b) Growth of data exchanged with size

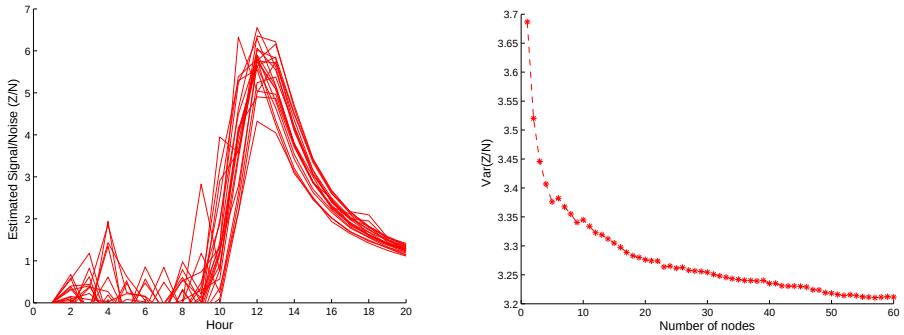


Figure 5: (a) Hourly variance of Z/N around an outbreak (b) Reduction in cumulative variance of Z/N as we add nodes

**Simulating background noise:** Our model for the background traffic is a Pareto distribution, a commonly used heavy tailed distribution whose probability density function is given as follows:

$$P(x) = \alpha b^{\alpha} x^{-(\alpha+1)} \tag{5}$$

A noteworthy aspect is that for $\alpha < 2$ heavy tailed distributions, such as the Pareto distribution, have infinite variance. Here $\alpha$ is referred to as the shape parameter and $b$ is the location parameter. Figure 3 provides an example of the background and worm traffic generated at twenty nodes for three different values of $K$, where $\alpha$ was fixed at $1.5$ and $b$ was fixed at $4.0$ for these experiments.

## 4.2 Data Filtering Strategies

We begin by describing three strategies for exchanging alert information in a distributed worm defense infrastructure such as DOMINO.

- **Scan Basis:** A simple strategy is to exchange complete header information of all scans that are observed in the network. This represents maximal filtering by sensor nodes. While such an approach can provide high fidelity alerts and the flexibility to perform sophisticated analysis, it might not be feasible due to privacy concerns or communication overhead.

- **Summary Basis:** An alternate strategy is to filter at sensors by summarizing the volume of activity seen at various ports. The centralized server could simply aggregate the volume summaries across all the sensors.[2] This simple strategy makes sense when you have diverse sensors providing logs from networks of various sizes.

[2]This aggregation model is similar to the one adopted by DSHIELD.org.

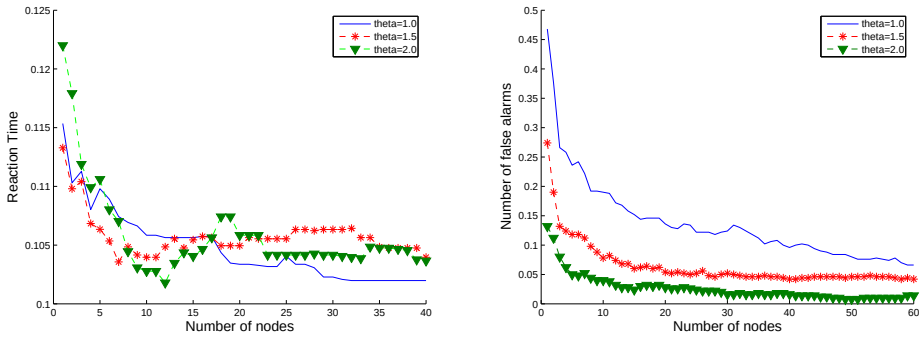


Figure 6: Reduction in (a) reaction time and (b) false alarms as we add nodes for three theta values

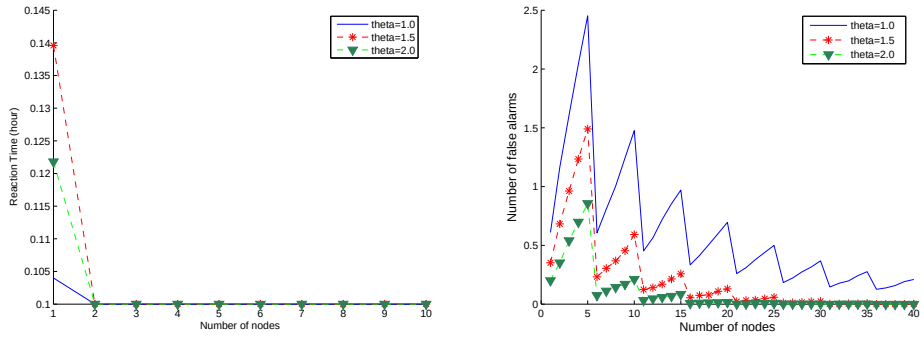


Figure 7: Reduction in (a) reaction time and (b) false alarms as we add nodes for three theta values using a voting function

- **Vote Basis:** The final strategy we consider is one in which sensors make a local decision and then simply pass alert/ no-alert data to the central server. The global decision function is then parametrized simply by the votes sent by the participating nodes. This strategy represents minimal filtering by sensor nodes.

## 4.3 Measuring Communication Overhead

We first evaluate the overhead of the three filtering strategies as a function of the number of data units that are exchanged under the three strategies. We assume that voting is trivial and hence set its weight to 1 data unit. Like voting, summaries grow linearly with the number of nodes, and are simply a constant size $C$ bigger than votes. We set $C$ to be 100. Finally, data exchange for scan basis grows both with the volume of traffic and the number of nodes. Here we assume each additional scan corresponds to one more data unit. The trade-off between data exchange and voting strategies is illustrated in Figure 4. The results, in some sense, can also be thought of as representing information loss due to filtering and thereby suggest compression methods as an interesting direction for future work.

## 4.4 Evaluating performance: Fusing Summaries

We first consider the performance implications of fusing summaries at a centralized server. Under this scenario, our fusion algorithm generates a single global alert if the signal to noise ratio $(\omega)$ exceeds a pre-defined threshold $(\theta)$. We estimate the $Z(t)$ and $N(t)$ from $X(t)$ as follows:

$$N(t) = X(t) - \frac{\sum_{i=1}^{t-1} X(i)}{t-1} \tag{6}$$

Now $Z(t)$ can be simply obtained by subtracting $N(t)$ from $X(t)$. Figure 5(a) shows the variance

of $(\omega)$ at each site during the 20 hour simulation period and provides an intuition for picking $\theta$. The fundamental trade-off here is between false alarms and reaction time. It seems that $\theta$ values greater than 1.5 might be resilient to false alarms. A different perspective is provided by Figure 5(b) that shows how aggregating data from more nodes reduces variance of $\omega$ and results in a more reliable signal. The impact of $\theta$ and additional sensors on reaction and time and false alarms is shown in Figure 6. This confirms our intuition from the previous graph: $\theta$ values between 1.5 and 2.0 provide reduced reaction time without many false alarms. The $\theta$ values chosen for the moving average detector might be dependent on the underlying variance and we are investigating other statistics such as the f-test that normalizes for the variance.

### 4.5 Evaluating performance: Fusing Votes

We perform a similar evaluation of the performance of the alert generation based on fusing votes. Each individual sensor generates an alarm using the same threshold $\theta$. A global alert is raised if at least 20% of the participants raise an alarm. Results for this strategy are shown in Figure 7.

## 5 Conclusions

In this paper, we explored the problem of combining data from distributed network sensors with the goal of improving timeliness and accuracy in identifying attacks. The contribution of this work is in defining the problem space, creating an analytic framework for understanding the utility of distributed detection and demonstrating the utility of distributed detection in simulation with two simple threshold-based data fusion methods. Our analysis shows that combining data from distributed sensors reduces the noise caused by benign traffic while enhancing the signal caused by the attack traffic. Our simulations demonstrate that even simple summary data fusion and voting schemes can be effective in increasing timeliness of attack identification and in reducing false alarm rates. These results indicate that distributed detection methods have great promise for defending networks from malicious attacks.

We are in the process of extending this work in three ways. The first is in a deeper exploration of the analytic methods for fusing data. It seems clear that our simple threshold-based methods can be improved. The second area of investigation is in applying our results to empirical data that we are collecting in conjunction with the DOMINO project. The third area is to expand our analysis to include both information and temporal diversity.

## References


[1] J. Allen, A. Christie, W. Fithen, J. McHugh, J. Pickel, and E. Stoner. State of the practice of intrusion detection technologies. Technical Report CMU/SEI-99-TR-028, Software Engineering Institute, Carnegie Mellon University, January 2000.

[2] R. Anderson and A. Khattak. The use of information retrieval techniques for intrusion detection. In *Proceedings of First International Workshop on the Recent Advances in Intrusion Detection (RAID)*, September 1998.

[3] J. Balasubramaniyan, J. Garcia-Fernandez, D. Isacoff, E. Spafford, and D. Zamboni. An architecture for intrusion detection using autonomous agents. In *Proceedings of the 14th Annual Computer Security Applications Conference (ACSAC'98)*, December 1998.

[4] J. Baras, A. Cardenas, and V. Ramezani. On-line Detection of Distributed Attacks from Space-time Network Flow Patterns. In *To appear in Proceedings of 24th Army Science Conference*, November, 2004.

[5] P. Barford, J. Kline, D. Plonka, and A. Ron. A signal analysis of network traffic anomalies. In *Proceedings of the ACM Internet Measurement Workshop*, Marseilles, France, November 2002.

[6] M. Barkat and P. Varshney. Decentralized CFAR Signal Detection. *IEEE Transactions on Aerospace and Electronic Systems*, 25(2), March 1989.

[7] W. Gong C. Zou and D. Towsley. Code red worm propagation modeling and analysis. In *Proceedings of the 9th ACM conference on Computer and Communications Security CCS'02*, Washington D.C., October 2002.

[8] R. Cáceres. Measurements of wide-area Internet traffic. Technical Report UCB/CSD 89/550, Computer Science Department, University of California, Berkeley, 1989.

[9] J. Chamberland and V. Verravalli. Decentralized detection in sensor networks. *IEEE Transactions on Signal Processing*, 51(2), February 2003.

[10] Symantec Corporation. Deepsight. http://www.enterprisesecurity.symantec.com, 2004.

[11] F. Cuppens and A. Miege. Alert correlation in a cooperative intrusion detection framework. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2002.

[12] R. Janakiraman, M. Waldvogel, and Q. Zhang. Indra: A peer-to-peer approach to network intrusion detection and prevention. *Unpublished Manuscript*, 2003.

[13] W. Lee, S.J. Stolfo, and K.W. Mok. A data mining framework for building intrusion detection models. In *Proceedings of the IEEE Symposium on Security and Privacy*, 1999.

[14] W. Leland, M. Taqqu, W. Willinger, and D. Wilson. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, pages 2:1–15, 1994.

[15] T. Lunt, A. Tamaru, F. Gilham, R. Jagannathan, P. Neumann, H. Javitz, A. Valdes, and T. garvey. A real-time intrusion detection expert system (IDES)-final technical report. Technical report, Computer Science Laboratory, SRI international, Menlo Park, CA, February 1992.

[16] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. Inside the slammer worm. In *Proceedings of the IEEE Security and Privacy Symposium*, Oakland, CA, July 2003.

[17] D. Moore, C. Shannon, and K. Claffy. Code red: A case study on the spread and victims of an internet worm. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, November 2002.

[18] D. Moore, C. Shannon, G. Voelker, and S. Savage. Internet quarantine: Requirements for containing self-propagating code. In *Proceedings of IEEE INFOCOM*, April 2003.

[19] D. Moore, G. Voelker, and S. Savage. Inferring internet denial of service activity. In *Proceedings of the 2001 USENIX Security Symposium*, Washington D.C., August 2001.

[20] S. Northcutt. *Network Intrusion Detection: An Analyst's Handbook*. New Riders, 1999.

[21] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson. Characteristics of Internet Background Radiation. In *Proceedings of ACM Internet Measurement Conference*, October, 2004.

[22] V. Paxson. *Measurements and Analysis of End-to-End Internet Dynamics*. PhD thesis, University of California Berkeley, 1997.

[23] V. Paxson. Bro: A system for detecting network intruders in real-time. *IEEE Computer Networks*, 31(23-24):2435–2463, 1999.

[24] M. Roesch. Snort- Lightweight Intrusion Detection for Networks. In *Proceedings of the 1999 USENIX LISA Conference*, November 1999.

[25] S. Snapp, J. Brentano, G. Dias, T. Goan, L. Heberlein, C. Ho, K. Levitt, B. Mukherjee, S. Smaha, T. Grance, D. Teal, and D. Mansur. DIDS (distributed intrusion detection system), - motivation, architecture, and an early prototype. In *Proceedings of the 14th National Computer Security Conference*, Washington, DC, 1991.

[26] S. Staniford, J. Hoagland, and J. McAlerney. Practical automated detection of stealthy portscans. In *Proceedings of the ACM CCS IDS Workshop*, November 2000.

[27] S. Staniford, V. Paxson, and N. Weaver. How to Own the Internet in Your Spare Time. In *Proceedings of the 11th USENIX Security Symposium*, August 2002.

[28] H.S. Teng, K. Chen, and S. C-Y Lu. Adaptive real-time anomaly detection using inductively generated sequential patterns. In *Proceedings of the IEEE Symposium on Security and Privacy*, 1999.

[29] M. Thottan and C. Ji. Anomaly detection in IP networks. *IEEE Transactions on Signal Processing*, 51(8), August 2003.

[30] J. Ullrich. Dshield.org. http://www.dshield.org, 2004.

[31] A. Valdes and K. Skinner. Probabilistic alert correlation. In *Proceedings of the International Workshop on the Recent Advances in Intrusion Detection (RAID'01)*, September 2001.

[32] H. Wang, D. Zhang, and K. Shin. Detecting SYN Flooding Attacks. In *Proceedings of IEEE INFOCOM*, April, 2002.

[33] C. Warrender, S. Forrest, and B. Pearlmutter. Detecting intrusions using system calls: Alternative data models. In *Proceedings of the IEEE Symposium on Security and Privacy*, 1999.

[34] V. Yegneswaran, P. Barford, and S. Jha. Global Intrusion Detection in the DOMINO Overlay System. In *Proceedings of Network and Distributed System Security Symposium (NDSS)*, February, 2004.

[35] V. Yegneswaran, P. Barford, and J. Ullrich. Internet intrusions: Global characteristics and prevalence. In *Proceedings of ACM SIGMETRICS*, San Diego, CA, June 2003.