Detection of Slow Malicious Worms using Multi-sensor Data Fusion

Frank Akujobi Ioannis Lambadaris Evangelos Kranakis

Abstract— **Detection of slow worms is particularly challenging due to the stealthy nature of their propagation techniques and their ability to blend with normal traffic patterns. In this paper, we propose a distributed detection approach based on the Generalized Evidence Processing (GEP) theory, a sensor integration and data fusion technique. With GEP theory, evidence collected by distributed detectors determine the probability associated with a detection decision under a hypothesis. The collected evidence is combined to arrive at an optimal fused detection decision by minimizing a cummulative decision risk function. Typically, malicious traffic flows of varying scanning rates can occur in the wild, and the difficulty in detecting slow scanning worms in particular can be exacerbated by interference from other traffic flows scanning at faster rates. Our proposed detection technique uses a window-based self adapting profiler to filter detected malicious traffic profiles with scanning rates greater than the low scanning rates we are interested in. Experiments on a live test-bed are used to demonstrate behavior of the technique.**

Keywords – Worms, Anomaly detection, Data fusion, Optimal decision.

I. Introduction and Related Work

Slow scanning malicious worms that blend with normal traffic patterns and evade intrusion detection systems (IDS) that depend only on anomalous network heuristics for detection have become an interesting research subject. Unlike fast scanning worms, this class of worms propagate through the network at rates below detection thresholds of network based intrusion detection systems and host based detection systems that use a number of incoming or outgoing connections as a basis for anomalous detection. Such worms are indistinguishable from normal traffic seen on the network, or seen by the end host network connections and are difficult to detect. Malicious slow scanning worms therefore pose a serious threat to networks today.

Some recent works have attempted to address the problem of detecting slow worms. In [1] a distributed end host detection scheme which uses a dynamic Bayesian network model for probabilistic detection was proposed for detection of slow worms. End host detectors alert when the number of outgoing connections to unique destination addresses and ports exceed a threshold and share detection information with each other to improve false detection rates. In [2] the SWORD detection system was proposed to detect zero-day worms of different propagation types and speeds. This was achieved by determining whether the total number of outgoing worm-like connections from a domain during a sliding window crosses a threshold set based on observation of normal traffic. However, it was acknowledged that if the worm speed is slow enough to cause interspersed traffic throughout a large amount of normal traffic, detection with the SWORD system becomes difficult. In [3] a multi-resolution approach for worm detection was proposed to deal with the limitations of simple threshold-based detection methods. Using a number of unique destinations contacted as a basis for anomaly detection, the multi-resolution approach used different thresholds during different time windows to detect attacks of different speeds. Faster scanning attacks were detected with smaller time windows while slower attacks were detected with larger time windows.

We point out that a common characteristic of most schemes proposed for detection of slow worms is the use of connection counts and traffic rates as the basis for anomalous detection. This approach inherently carries a high rate of false alarms because slow worms are capable of propagating at rates similar or less than normal traffic rates and therefore can camouflage as normal traffic. Also, information about vulnerabilities and attempted exploits do not exist in the network layer [4], hence such techniques are unable to provide verifiable evidence of malicious intrusions. In fact, the assumption that malicious attacks necessarily cause anomalous activity in the network in terms of host or network traffic was recently challenged in [5].

On the other hand, host-based Anomaly Intrusion Detection Systems (AIDS) which infer suspicious activity when detector endpoints experience an intrusion that attempts to alter a pre-defined standard state [1] of the endpoint have been more successful at detecting malicious worm intrusions irrespective of scanning behavior of worms. Typically, such attempts are in the form of anomalous system calls [6], unauthorized or infectious intrusions which cause the host AIDS to trigger an alert. Recent work [4] and vendor implementations [7] have recorded success in using host AIDS for detecting unauthorized intrusions. Host AIDS are capable of leveraging large amounts of detailed context about applications and system behavior to effectively detect anomalous host behaviors [8]. The technique adopted in [4] shows that with properly instrumented detection software, host-based intrusion detection is effective and capable of minimizing false positives.

In this paper, we use the Generalized Evidence Processing (GEP) theory, a multi-sensor data fusion technique, for combining intrusion detection evidence provided by distributed host-based intrusion detectors [9] [10]. There has been previous attempts to use two major evidence combining theories for intrusion detection - the Bayesian theory and the Dempster-Shafer theory [11] [12]. Proponents of the Bayesian theory criticize the Dempster-Shafer theory for lack of rigorousness in the axiomatic definition of evidence and the inability to use a priori probabilities when they are known [9] [13]. On the other hand, proponents of the Dempster-Shafer theory criticize the Bayesian theory for lack of flexibility when it comes to fuzzy decisions where the evidence might not support hard decisions, difficulty in defining a priori probabilities and likelihood

Frank Akujobi and Ioannis Lambadaris are with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada (email: fakujobi, ioannis@sce.carleton.ca).

Evangelos Kranakis is with the Department of Computer Science, Carleton University, Ottawa, ON, Canada (email: kranakis@scs.carleton.ca).

This research was partially funded by grants from Natural Sciences and Engineering Research Council (NSERC) and Mathematics of Information Technology and Complex Systems (MITACS).

[1] Pre-defined standard states of endpoints are typically determined by established security policies and standards.

functions, as well as the mutual exclusivity requirement for competing hypotheses [12] [14]. The GEP theory unifies both theories in a generalized framework and combines their advantages [9] [14] [13]. With GEP theory, the evidence collected by the host detectors determines the probability associated with a decision under a hypothesis. The probability assignments may be based on the Bayesian likelihood function or correspond to the belief functions used in the Dempster-Shafer evidential theory. The evidence is combined to arrive at an optimal fused decision by minimizing a cummulative risk function.

In [15] we presented a Bayesian inference technique for detecting both fast and slow scanning worms. Though functional, this technique inherits all the limitations of the Bayesian framework - mutual exclusivity requirement for competing hypotheses and lack of support for indecision. Also, even though in [15] we computed the a priori probability of an unauthorized intrusion event from gathered data, a more accurate approach would have been to determine a priori probability before data gathering. Determination of a priori probabilities when they are not known is another difficulty that limits the practical use of the Bayesian technique. The GEP theory addresses these shortcomings and provides an optimal way of combining evidence to arrive at a decision.

We emphasize that slow worms do not exist alone in the wild. Typically, malicious intrusion traffic of varying scanning rates coexist in the wild and their existence can introduce false alarms to detection of slow worms. Our detection approach adaptively filters traffic profiles with scanning rates greater than the low rates we are interested in and uses an optimized detection technique for slow worm detection.

A. Contributions

The main contributions of this work are:

- We propose an optimized intrusion detection scheme based on the Generalized Evidence Processing theory, a sensor integration and data fusion technique known to have advantages over the two major evidence combining theories that have dominated the field of distributed evidence processing - the Bayesian theory and the Dempster-Shafer theory.
- Our algorithm takes into consideration the real possibility that faster propagating malicious intrusions can co-exist with slow worms in computer networks, and therefore interfere with slow worm detection.
- We use a combination of evidence from host-based anomaly detectors, a detection window-based profiler and GEP-based data fusion for detection of slow worms.
- Experimenting on a live test-bed we demonstrate the technique and present results.

B. Outline

Section II introduces the Generalized Evidence Processing theory. In Section III we describe the proposed detection technique for slow worms. Experimentation on a live test-bed with the proposed technique is presented in section IV and in section V we conclude the paper and point to future work.

II. Introduction - Generalized Evidence Processing Theory

Some known limitations of the classical and Bayesian decision processes include inability to deal with both non-mutually exclusive multiple hypotheses and uncertainty [12] [14]. The GEP theory extends the Bayesian inference framework to deal with these limitations. As an example, let H_0 and H_1 be the two hypotheses under testing. The events associated with the probability space can be attributed to the two hypotheses H_0 and H_1 with probabilities $P(H_0) \geq 0$ and $P(H_1) \geq 0$ respectively, where $P(H_0)+P(H_1) = 1$. Let D_0, D_1, and D_2 be the decisions which correspond to the propositions "H_0 is true", "H_1 is true" and "H_0 or H_1 is true" [2] respectively. With classical and Bayesian inference where H_0 and H_1 are mutually exclusive events, the probability associated with D_2 is equivalent to:

$$P(D_2) = P(H_0 \cup H_1) = P(H_0) + P(H_1) = 1$$

This shows an inability to account for non-mutually exclusive events and uncertainty (or indecision) within the Bayesian framework. The GEP theory is a unified evidence theory which accounts for indecision and combines evidence that supports non-mutually exclusive propositions to arrive at a decision by minimizing a cummulative risk function. In a distributed multi-sensor system with N sensors, let $\mathbf{Z}$ be the observation (data) space which results in individual local decisions on the sensors. We represent $\mathbf{Z}$ as $\mathbf{Z} = \{\underline{z} : \underline{z} = (z_1, z_2, .., z_N), z_i = 0, 1, 2\}$ where 0 implies a "benign observation", 1 implies a "malicious observation" and 2 implies an "uncertainty" about the nature of the observation. Also, let two hypotheses H_1 and H_0 be considered, where H_1 is the hypothesis that the observation is malicious and H_0 is the hypothesis that the observation is benign. Each local sensor observation results in a local sensor decision (see Fig. 1). Hence, the vector of observations $\underline{z}$ results in a vector of local decisions $\{\underline{d} : \underline{d} = (d_1, d_2, .., d_N), d_i = 0, 1, 2\}$, where 0, 1, and 2 are the individual local decisions which correspond to the propositions "H_0 is true", "H_1 is true" and "H_0 or H_1 is true" (i.e. an indecision)

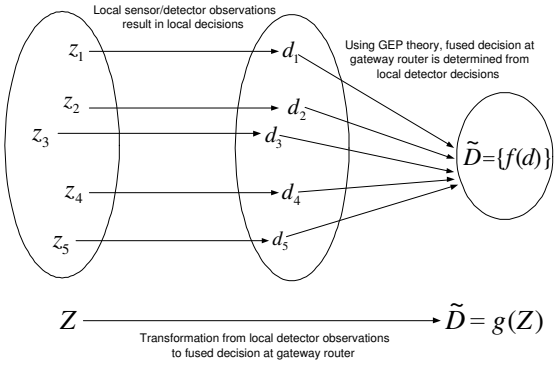


Fig. 1. Transformation from N local detector observations to a fused decision. $N = 5$.

Using the GEP theory, the local sensor decisions are combined at a fusion center to arrive at a fused decision that minimizes a cummulative risk function. As depicted in Fig. 1, let $g(\mathbf{Z})$ be the transformation from the observation space $\mathbf{Z}$ into the fused decision

[2] Decision D_2 therefore represents an indecision about the true nature of the hypothesis.

space $\tilde{\mathbf{D}} = \{\mathbf{D} = f(\underline{d}) : \underline{d} = (d_1, d_2, .., d_N), d_i = 0, 1, 2\}$ such that,

$$\tilde{\mathbf{D}} = g(\mathbf{Z}), \quad \tilde{\mathbf{D}} = \{0, 1, 2\} \quad \text{and} \quad \mathbf{D} = 0, 1, 2 \tag{1}$$

where $\mathbf{D} = 0, 1, 2$ are the fused decisions that "H_0 is true", "H_1 is true" and "H_0 or H_1 is true" respectively.

In practical worm detection systems, distributed detectors in a network can make observations of malicious intrusions in the network and report their individual decisions to a central processor (such as a gateway router). The transformation $g(\mathbf{Z})$ corresponds to the function of a correlation algorithm running on the gateway router that takes as input the individual local decisions of the detectors and outputs a fused decision.

Following the GEP theory [9], let C_{ab} be the cost associated with a decision a that is in set $\tilde{\mathbf{D}}$ at the fusion center when hypothesis H_b is true, where $0 \leq C_{ab} \leq 1$. We assume that there is no penalty for a correct decision, hence the associated cost for a correct decision is zero (i.e. $C_{00} = C_{11} = 0$). It can be shown (see Appendix I) that the decision arrived at in (1) can be optimally determined by minimizing the cummulative risk $\mathbf{R}$ at the fusion center using the following decision rules:

$$\Lambda(\underline{d}) \underset{\mathbf{D}=0 \text{ or } \mathbf{D}=2}{\overset{\mathbf{D}=1 \text{ or } \mathbf{D}=2}{\gtrless}} \frac{P(H_0)}{P(H_1)} \frac{C_{10}}{C_{01}} \tag{2}$$

$$\Lambda(\underline{d}) \underset{\mathbf{D}=0 \text{ or } \mathbf{D}=1}{\overset{\mathbf{D}=2 \text{ or } \mathbf{D}=1}{\gtrless}} \frac{P(H_0)}{P(H_1)} \frac{C_{20}}{C_{01} - C_{21}} \tag{3}$$

$$\Lambda(\underline{d}) \underset{\mathbf{D}=2 \text{ or } \mathbf{D}=0}{\overset{\mathbf{D}=1 \text{ or } \mathbf{D}=0}{\gtrless}} \frac{P(H_0)}{P(H_1)} \frac{C_{10} - C_{20}}{C_{21}} \tag{4}$$

where,

$$L \underset{d_y}{\overset{d_x}{\gtrless}} R$$

implies that decision d_x is made if $L > R$, otherwise decision d_y is made.

According to equation (2), (3), (4), the fusion decision rules depend on the values of the C_{ab} costs and a priori probabilities $P(H_1)$ and $P(H_0)$ of the two hypotheses, H_1 and H_0 respectively. We assume that the a priori probabilities are known, hence we are interested in estimating $\Lambda(\underline{d})$.

For malicious worm detection, C_{10} and C_{01} are the costs associated with a false positive decision and a false negative decision respectively. Slow scanning worms are known to exhibit high rates of false negatives since they are capable of avoiding detection by scanning at rates below most traditional IDS thresholds and blending with normal traffic patterns. As a result, unlike fast worms they inherently exhibit greater false negative rates than false positive rates. Slow worm detection therefore has a higher risk of false negatives, hence we use cost values $C_{01} > C_{10}$ for slow worm detection which ensures a greater penalty for false negatives than false positives. For worm detection systems without a bias for worm speed, cost values $C_{10} = C_{01}$ is appropriate to ensure the same penalty for decisions that result in either false positives or false negatives.

Equation (2), (3), (4) also show that the GEP framework can make use of the a priori probabilities of both hypothesis H_1 and H_0 if they are known. When they are not known, we assume that $P(H_1) = P(H_0)$ thus nullifying the impact of a priori probabilities on the fusion decision rules in (2), (3), (4). Also, note that the GEP decision process breaks down to a binary decision process if *indecision* is not considered [3].

To illustrate an application of the decision rules, we consider different possible cases as was done in [9]. We assume a priori probabilities of both hypothesis H_1 and H_0 are unknown, hence $P(H_1) = P(H_0)$, and that the cost of an incorrect decision is greater than the cost associated with an indecision (i.e. $C_{10} > C_{20}$, $C_{01} > C_{21}$).

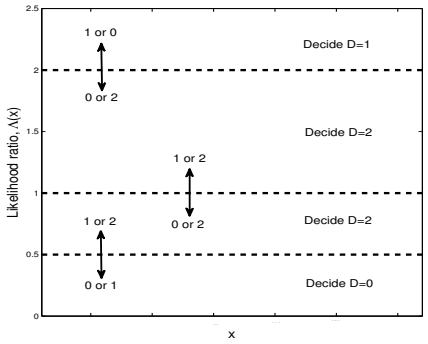


Fig. 2. Case 1: The indecision region lies between the two definite decision regions.

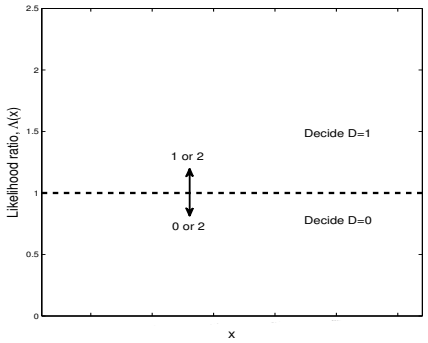


Fig. 3. Case 2: The indecision region is completely eliminated.

A. *Case* 1: $C_{00} = C_{11} = 0$, $C_{10} > 2 * C_{20}$, $C_{01} > 2 * C_{21}$

Let $C_{10} = C_{01} = 1$, $C_{20} = C_{21} = \frac{1}{3}$. Hence, $\frac{C_{10}}{C_{01}} = 1$, $\frac{C_{20}}{C_{01} - C_{21}} = 0.5$, and $\frac{C_{10} - C_{20}}{C_{21}} = 2$. Equation (2), (3), (4) are used to generate the partition in Fig. 2. In this case, Fig. 2 shows that the indecision region lies between the two definite decision regions. This case is applicable to practical detection systems that are not always capable of providing evidence to support definite decisions, hence the option of indecision is provided.

[3] In this case, only the decision rule (2) applies since C_{21} and C_{20} become undefined.

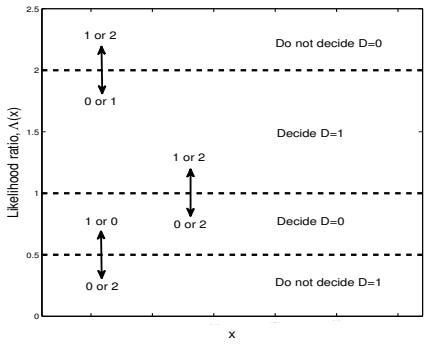


Fig. 4. Case 3: The definite decision regions lies between two indecision regions.

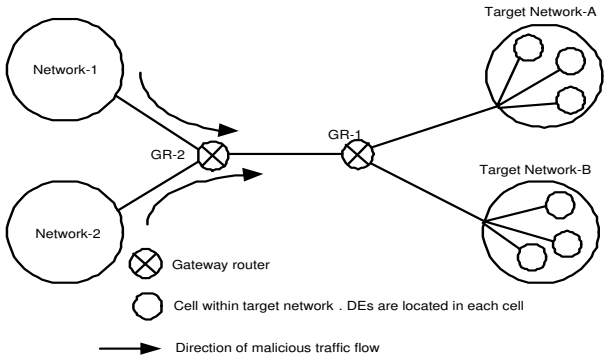


Fig. 5. Typical worm attack on multiple networks

B. *Case* 2: $C_{00} = C_{11} = 0, C_{10} > 2 * C_{20}, C_{01} > 2 * C_{21}$

Let $C_{10} = C_{01} = 1$, $C_{20} = C_{21} = 0.5$. Hence, $\frac{C_{10}}{C_{01}} = \frac{C_{20}}{C_{01}-C_{21}} = \frac{C_{10}-C_{20}}{C_{21}} = 1$. All three thresholds have the same values and the indecision region is non-existent as shown in Fig. 3. In this case the decision process corresponds to a standard binary decision process. This case is applicable if the detection system is capable of always providing hard evidence sufficient to support a decision or if the system is not capable of dealing with indecision.

C. *Case* 3: $C_{00} = C_{11} = 0, C_{10} > 2 * C_{20}, C_{01} > 2 * C_{21}$

Let $C_{10} = C_{01} = 1$, $C_{20} = C_{21} = \frac{2}{3}$. Hence, $\frac{C_{10}}{C_{01}} = 1$, $\frac{C_{20}}{C_{01}-C_{21}} = 2$, and $\frac{C_{10}-C_{20}}{C_{21}} = 0.5$. In this case, Fig. 4 shows that the two definite decision regions lie between two indecision regions, an exact opposite of Case 1. Case 3 represents a detection system that exhibits a standard binary decision process within a likelihood ratio bound (in this case $0.5 < \Lambda(x) < 2$). Beyond the bound, the detection system is incapable of making a definite decision.

Practical detection systems are more suited to Case 1 and Case 2.

III. Proposed Detection Approach

Fig. 5 depicts a typical worm intrusion scenario in which attackers in Network-1 and Network-2 launch scanning worm attacks on Network-A and Network-B. Typically, well-designed enterprise networks are logically subdivided into cells or network zones as shown in Fig. 5. The detection scheme uses detector endpoints within distributed cells in a target network for detection of intrusion attempts and combines the observed intrusion data on the gateway router of the cells.

TABLE I
Detection Parameters

Notation	Explanation
SW_j	j^{th} slow worm detection window
W_{kj}	k^{th} worm detection window within SW_j
t_s	duration of slow worm detection window
t_w	duration of worm detection window
Z_j	set of profiles captured by the SWDA during SW_j
X_{kj}	worm profiles detected during the W_{kj} window
Y_j	set of profiles forwarded to the slow worm correlation engine (SWCE)

A. *Detection Technique*

Our technique uses two instances of detector agents, worm detector agent (WDA) and slow worm detector agent (SWDA). Both run simultaneously on hardened detector endpoints (DEs) located within distributed cells in the network and are responsible for capturing malicious intrusion attempts targeted at the cells. We assume in this work that the detector agents run host-based anomaly detection software configured to capture intrusion data when malicious intrusions are detected on the detector endpoints. While the SWDA is used for detection of slow propagating malicious worms, the WDA is used for detecting worm intrusions that are not necessarily slowly propagating. This is achieved by capturing intrusion data during two different detection time window intervals. Table I and Fig. 6 describe some of the detection algorithm parameters.

1) Intrusion Detection Windows (W_{kj}, SW_j): We refer to an epoch that spans a capture interval as a detection window. Two detection windows are used in our detection algorithm - the "worm detection window" and the "slow worm detection window". A *worm detection window* refers to an epoch of duration t_w started as a result of an intrusion attempt detected by a WDA. A *slow worm detection window* refers to a periodic epoch of duration t_s which runs continuously on each SWDA. Fig. 6 shows a snapshot of a series of epochs during which the WDA and SWDA carry out real-time recording of network traffic profiles. Typically, $t_s > t_w$, hence there could be multiple worm detection windows within a single slow worm detection window (Fig. 6). At the end of a worm detection window, all profiles recorded by a WDA running on a DE in the cell are transferred to the first upstream gateway router for correlation. We define a *profile* as a 4-tuple consisting of *srcIP*, *dstport*, *proto*, *payload*. *srcIP* is the source IP address in the IP header of packets captured by the DE, *dstport* is the target port, *proto* is the transport layer protocol used and *payload* is the signature of the exploit in the payload of the IP packet. After the transfer, the WDA continues to monitor for future intrusion attempts. The SWDAs wait until the end of the slow worm detection window before transfering captured records to the gateway router. The next slow worm detection window is started immediately after the transfer.

2) Worm Detection: When a WDA running on a detector endpoint (DE) makes a positive detection of a malicious or unauthorized intrusion the following occurs:

- The WDA immediately sends an alert to other participating WDAs respectively within the cell. WDAs communicate only

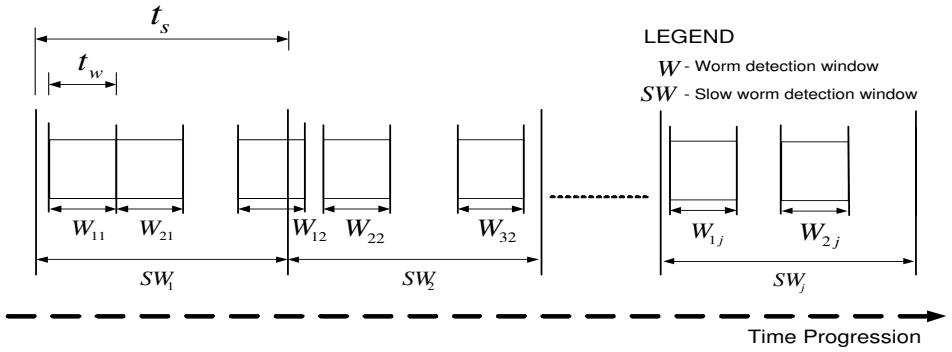


Fig. 6. Series of epochs showing detection windows

with other WDAs.

- When the alert is received, the WDAs within the target cell start real-time recording of *profiles* for all network traffic originated from outside their cell and targeted at the DEs for a preset capture interval. The WDA capture interval corresponds to the *worm detection window* with duration t_w.
- For each traffic profile i detected in the target cell by a WDA, two hypotheses H_1 and H_0 are considered, where H_1 is the hypothesis that the traffic profile i is malicious and H_0 is the hypothesis that the traffic profile i is benign. For the profile i, let d_q^i be the individual local decision by the WDA on the q^{th} DE based on observed intrusion attempts. $d_q^i = 1$ if H_1 is decided and $d_q^i = 0$ if H_0 is decided. We assume the anomaly detection software running on a WDA is capable of making such a decision. For this work, we considered a binary local detection outcome which did not include indecision. However, the GEP theory is capable of dealing with indecision as explained earlier. For a target cell with m DEs, let $\underline{d}^i = (d_1^i, d_2^i, .., d_m^i)$ be the vector of individual WDA decisions on traffic profile i.
- At the end of the worm detection window, the WDAs on all DEs in the cell transfer their records and local decisions to their upstream gateway router and continue monitoring the DEs for unauthorized intrusions.

3) Slow Worm Detection: As mentioned earlier, we assume both SWDA and WDA use the same anomaly detection mechanism, though the SWDA records network traffic profiles for a longer period, t_s. The SWDAs perform continuous real-time capturing of *profiles* of all network traffic originated from outside their cell and targeted at the DEs in epochs of interval t_s which corresponds to the *slow worm detection window*. During a slow worm detection window, if an SWDA running on a DE detects a malicious or unauthorized intrusion attempt it captures the nature of the attempted intrusion and continues real-time recording of incoming traffic profiles. The capture reveals useful information about a possible exploit and vulnerability on hosts in the cell. At the end of a slow worm detection window, the SWDAs on all DEs in the cell transfer their records and local decisions to their first upstream gateway router and immediately start the next epoch of recording. Unlike the WDAs, the SWDAs do not wait for an alert before capturing intrusion data. Intrusion data is captured in periodic slow worm detection windows of duration t_s.

The WDA and SWDA on the DEs do not initiate communication with any host outside their cell nor do they participate in normal traffic transactions. Their role is to make local decisions (*malicious* or *benign*) concerning detected intrusions and communicate that decision to their gateway router.

B. Correlation Technique

The upstream gateway router receives the records and local decisions transfered from the WDAs and SWDAs on DEs in the target cell. The gateway router runs two correlation engines, worm correlation engine (WCE) which executes a worm correlation algorithm (WCA) and a slow worm correlation engine (SWCE) which executes a slow worm correlation algorithm (SWCA). Both WCA and SWCA use the GEP evidence combining technique to determine the most likely profile(s) associated with the detected malicious or unauthorized intrusion(s). Multiple correlation processes can run on the gateway router simultaneously.

1) Worm Correlation Algorithm (WCA): At the gateway router, we are interested in using collected WDA local decisions in making an optimal fused decision which minimizes a cummulative decision risk. For each traffic profile i with associated records and local decisions received from the WDAs, two hypotheses H_1 and H_0 are considered, where H_1 is the hypothesis that the traffic profile i is malicious and H_0 is the hypothesis that the traffic profile i is benign. In general, $P_i(H_1)$ and $P_i(H_0)$ can be estimated using historical data or experience. However, without loss of generality, we assume that the a priori probabilities of the two hypothesis $P_i(H_1)$ and $P_i(H_0)$ for each profile i are the same. Hence, the GEP optimal decision criteria at the fusion centre (the gateway router) which minimizes the cummulative decision risk can be expressed using the following likelihood ratio rule (derived from (2)):

$$\Lambda(\underline{d}^i) = \frac{P(\underline{d}^i|H_1)}{P(\underline{d}^i|H_0)} \underset{\mathbf{D}=0}{\overset{\mathbf{D}=1}{\gtrless}} \frac{C_{10}}{C_{01}} = \gamma \tag{5}$$

where C_{10} and C_{01} are the costs of a false positive decision and a false negative decision respectively. The choice of C_{10} and C_{01} is system design driven and in our system implementation we used $C_{10} = C_{01}$, which ensures the same penalty for both a false positive and a false negative decision, thus exhibiting no bias for the speed of the worm. With our implementation, the GEP decision process breaks down to a binary decision process, hence we do not consider indecision [4]. This corresponds to Case 2 in Section II.

[4]Indecision within the GEP theory framework is reserved for future work.

TABLE II

PARAMETERS FOR GEP-BASED CORRELATION ALGORITHM

Notation	Explanation
P_{Di}	Combined probability of positive detection for traffic profile i
P_{Fi}	Combined probability of false detection for traffic profile i
p_{dq}	Detection probability for the q^{th} individual detector
p_{fq}	False alarm probability for the q^{th} individual detector
d_q^i	Individual local binary decision by the q^{th} DE on intrusion attempts due to profile i.
$\Lambda(\underline{d}^i)$	GEP likelihood ratio for optimal fused decision
γ	GEP likelihood ratio threshold, also equivalent to $\frac{C_{10}}{C_{01}}$
C_{ab}	Cost or penalty associated with a detector decision a when the true hypothesis is H_b
m_i	Total number of detectors with observations of profile i
u_i	Total number of detectors with observations of profile i and that favor H_1
v_i	Number of detectors which favor H_1 required to minimally satisfy $\Lambda(\underline{d}^i) \geq \gamma$

To express (5) in more practical terms, let p_{dq} denote the detection probability and p_{fq} denote the false alarm probability of the q^{th} individual detector. Both p_{dq} and p_{fq} depend on the quality of the detector. In our implementation, the WDAs are homogeneous [5] since all DEs are assumed to run the same anomaly host-based intrusion detection software, hence $p_{dq} = p_d$ and $p_{fq} = p_f$, for all q. Also, $p_d > p_f$. See Table III for the relationship between H_1, H_0 and p_d, p_f. For a particular profile i, let m_i be the total number of

TABLE III

RELATIONSHIP BETWEEN H_1, H_0 AND p_d, p_f

True Nature	Detector decision	
	H_1	H_0
H_1	p_d	$1-p_d$
H_0	p_f	$1-p_f$

detectors in the target cell with observations of profile i and u_i be the total number of such detectors with individual local decisions which favor H_1. If we assume the observations on individual DEs are conditionally independent given hypotheses H_1 and H_0, then according to GEP, the conditional probability at the gateway router is

$$P(\underline{d}^i|H_1) = p_d^{u_i} * (1-p_d)^{m_i-u_i}$$

$$P(\underline{d}^i|H_0) = p_f^{u_i} * (1-p_f)^{m_i-u_i}$$

Hence, the likelihood ratio test in (5) is equivalent to,

$$\Lambda(\underline{d}^i) = \frac{P(\underline{d}^i|H_1)}{P(\underline{d}^i|H_0)} = \left(\frac{p_d}{p_f}\right)^{u_i} * \left(\frac{1-p_d}{1-p_f}\right)^{m_i-u_i} \underset{\mathbf{D}=0}{\overset{\mathbf{D}=1}{\gtrless}} \frac{C_{10}}{C_{01}} = \gamma \quad (6)$$

Based on computation of $\Lambda(\underline{d}^i)$, the likelihood ratio test in (6) determines whether the correlation algorithm considers traffic profile i as a malicious traffic profile (i.e. $\mathbf{D} = 1$) or a benign traffic profile (i.e. $\mathbf{D} = 0$).

2) Threshold-based selection for worm containment: To detect slow scanning worms, we first identify and filter (or contain) faster scanning malicious traffic profiles if they exist. Our proposed approach involves computing the combined probability of detection, P_{Di} and the combined probability of false detection, P_{Fi} for each traffic profile i determined to be malicious using the likelihood ratio test in (6). Then, a detection probability threshold parameter Φ_D and a false detection probability threshold parameter Φ_F are used to select suspicious traffic profiles based on the significance of their combined detection probability and combined false detection probability.

P_{Di} and P_{Fi} for each detected traffic profile i are computed using the following expressions (see Table II for description of notations):

$$P_{Di} = P(\Lambda(\underline{d}^i) \geq \gamma|H_1) \quad (7)$$

$$P_{Fi} = P(\Lambda(\underline{d}^i) \geq \gamma|H_0) \quad (8)$$

If we let v_i be the minimum number of detectors which favor H_1 required to satisfy the condition $\Lambda(\underline{d}^i) \geq \gamma$ (determined using 6), then P_{Di} and P_{Fi} defined in (7) and (8) can be expressed as:

$$P_{Di} = \sum_{u_i:u_i \geq v_i} \binom{m_i}{u_i} (p_d)^{u_i} (1-p_d)^{m_i-u_i}$$

$$P_{Fi} = \sum_{u_i:u_i \geq v_i} \binom{m_i}{u_i} (p_f)^{u_i} (1-p_f)^{m_i-u_i}$$

For selection of faster malicious traffic profiles (if they exist), a threshold-based selection process follows for each profile i observed by detectors in the target cell.

```
For each profile i, if (P_Di > Φ_D) and (P_Fi < Φ_F),
  {
   then select profile i;
   trigger automated containment;
  }
else,
  {
   do not select profile i;
  }
```

In our experiment (section IV), tractable values for Φ_D and Φ_F were chosen to demonstrate the behavior of the proposed detection technique after observing several experiment runs. More work is required to develop an optimal technique for threshold determination.

[5] The implementation can be modified to use heterogeneous WDAs.

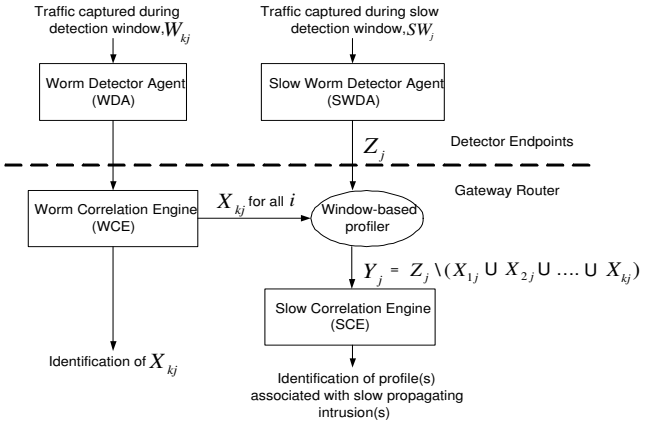


Fig. 7. Flow diagram of slow worm detection.

3) Window-based profiler: Let W_{kj} be the k^{th} worm detection window within the j^{th} slow worm detection window SW_j (see Table I and Fig. 6). We use X_{kj} to model a set with elements corresponding to profile(s) identified as associated with worm intrusion(s) propagating at rates greater than the slow scanning rates we are interested in. These profiles are identified by the WCE from records captured during W_{kj} (Fig. 7). We also use Z_j to model the set with elements corresponding to all profiles captured by the SWDAs in the cell during the slow worm detection window SW_j. As mentioned earlier, we assume the SWDAs run that same host anomaly detection software as the WDAs but record traffic profiles for a longer period, t_s which corresponds to the slow worm detection window SW_j. For each slow worm detection window, the profiler tags traffic profiles identified by the WCE and periodically adapts the input into the slow worm correlation engine (SWCE) by filtering out those profiles. This ensures that only profiles that have not been previously selected by the WCE as associated with faster propagating intrusions are forwarded to the SWCE (Fig. 7). A similar adaptive profiler technique was used in [15] to filter out traffic profiles belonging to fast scanning worms. If Y_j is the set with elements corresponding to profiles forwarded to the SWCE, then Y_j is expressed as:

$$Y_j = Z_j \setminus (X_{1j} \cup X_{2j} \cup ..... \cup X_{kj}) \tag{9}$$

This profiler algorithm ensures that for every slow worm detection window, SW_j, the corresponding Y_j is updated with outputs, X_{kj} from the WCE. At the end of a slow worm detection window, only profiles that are not deemed to belong to faster propagating intrusions by the WCE are forwarded to the SWCE for slow worm detection and identification. The SWCE runs the slow worm correlation algorithm (SWCA) described in the next section.

4) Slow Worm Correlation Algorithm (SWCA): The SWCE runs the slow worm correlation algorithm (SWCA) on Y_j. Slow scanning worms are known to exhibit high rates of false negatives since they are capable of avoiding detection by scanning at rates below most traditional IDS thresholds and blending with normal traffic patterns. As a result, unlike fast worms they inherently exhibit greater false negative rates than false positive rates. We use $\gamma_s = \frac{C^s_{10}}{C^s_{01}}$ where $0 \leq C^s_{ab} \leq 1$ to denote the decision cost ratio used for slow worm detection in the SWCA. We also use $\underline{d_s}^j$ to denote the vector of local decisions from the SWDAs corresponding to the elements in Y_j. The slow worm correlation algorithm (SWCA) detects slow worm profiles by using the following likelihood ratio rule:

$$\Lambda(\underline{d_s}^j) = \frac{P(\underline{d_s}^j|H_1)}{P(\underline{d_s}^j|H_0)} \underset{\mathbf{D}=0}{\overset{\mathbf{D}=1}{\gtrless}} \gamma_s = \frac{C^s_{10}}{C^s_{01}} \tag{10}$$

where $C^s_{01} > C^s_{10}$, thus ensuring a greater penalty for false negatives than false positives. The choice of γ_s determines the minimum number of positive detectors (i.e. SWDAs) required to satisfy (10). This technique can also be used for detecting stealthy worms that are not necessarily scanning worms as long as such worms infect at least the minimum number of hosts required to satisfy (10) within the SW_j window. In our experiment, we used $\gamma_s = 0.5$.

IV. Experimentation

A. Description of test-bed setup

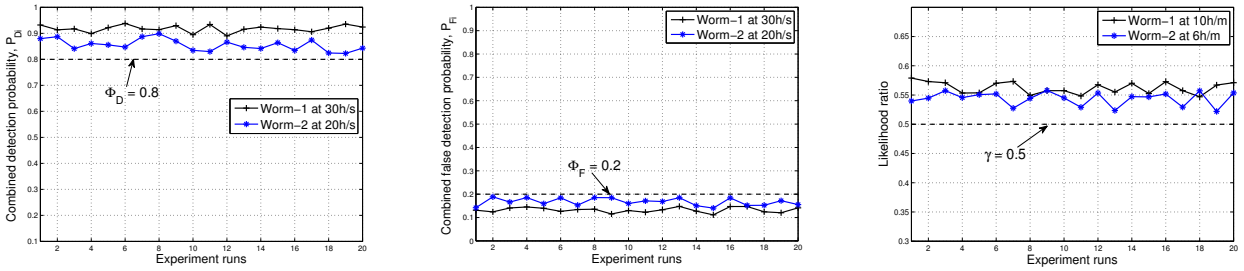
Fig. 5 shows the topology of our live testbed which will be described in more detail in this section. Worm attacks are sourced from Network-1 and Network-2 and targeted at vulnerable hosts in Network-A and Network-B. Network-A and Network-B are logically subdivided into cells or network zones as shown in Fig. 5. Detector endpoints (DEs) that run our detection algorithm are located within the target cells and communicate with their gateway router (GR-1). The gateway router runs our GEP-based correlation algorithm.

To evaluate the functionality and performance of our proposed detection scheme, we emulated self propagating slow worm attacks using a modified *blaster worm* source code [16]. To emulate multiple malicious attacks the source code was used to instrument two worms that exploited two different vulnerabilities. The first, *worm-1* was instrumented to create a directory named */root/infected-1* on the target host and copy a file named *malicious-1* into that directory over TCP port 888. The second, *worm-2* was instrumented to create a directory named */root/infected-2* on the target host and copy a file named *malicious-2* into that directory over UDP port 999. Hosts in Network-1 and Network-2 were used to launch *worm-1* and *worm-2* random attacks respectively on hosts in the target networks (Network A and Network B). Emulated slow worms with scanning rates of 6h/m and 10h/m were used in our experiment. Slow worm rates and thresholds in the order of this magnitude have been used in previous works [2] [1]. To demonstrate normal worm activity that co-exist with our slow worms we emulated worms with scanning rates of 20h/s and 30h/s. In comparison, the Witty worm [17] infected 110 hosts in the first 10 seconds, equivalent to an average infection rate of about 11h/s while the Slammer worm [18] infected more than $75,000$ hosts within 10 minutes, equivalent to an average infection rate of over 125h/s.

We used OpenVZ virtualization [6] [19] to create the required vulnerable host population in the target networks. Up to 64 virtual hosts per workstation were created on Linux workstations running OpenVZ kernel-2.6.22 to emulate a vulnerable population in each target network.

For our test, the host-based Anomaly Intrusion Detection System (AIDS) running on the worm detector agent (WDA) and the

[6]OpenVZ is an operating system-level virtualization technology based on the Linux kernel and operating system.



(a) Combined detection probability, P_{Di}. $\Phi_D = 0.8$.

(b) Combined false detection probability, P_{Fi}. $\Phi_F = 0.2$.

(c) Likelihood ratio $\Lambda(\underline{d_s}^j)$ for slow worm profiles. $\gamma_s = 0.5$.

Fig. 8. Experimental results. $m_i = 24$. $v_i = 16$. Number of hosts in cell = 128

slow worm detector agent (SWDA) were emulated using different instances of snort-based IDS that constantly monitored the directory structure and content of the DE, and generated an alert when a file named *malicious-1* or *malicious-2* was found in a directory named */root/infected-1* or */root/infected-2* respectively on the DE. We used a probability of detection, $p_d = 0.8$. In our implementation, the snort-based IDS was used for real-time recording on the WDA and SWDA [7]. The parameter t_w was set to 10 seconds on the WDA to ensure that the average number of hosts hit by emulated normal scanning worms (20h/s and 30h/s) used in our experiment exceeded v_i for each target network. The parameter t_s was set to 25 minutes on the SWDA to ensure that the slowest scanning worm rate of interest (6h/m) registered hits within the t_s window. The gateway router, GR-1 ran instances of our proposed worm correlation engine (WCE) and slow correlation engine (SWCE).

The purpose of the experiment was to demonstrate how the proposed detection technique can be used for detecting slow propagating worm attacks. It may not representative of all the possible worm attack scenarios that exist or may exist on the Internet today.

B. Description of experiment

In this experiment, four attacking hosts, two from Network-1 and two from Network-2 in Fig. 5 were used to launch different attacks (worm-1 and worm-2 respectively) on hosts in the target networks. The scanning rate of a pair of worm-1 and worm-2 attacks were set to 20h/s and 30h/s respectively to emulate normal scanning worms. The scanning rate of the second pair of worm-1 and worm-2 attacks were set to 6h/m and 10h/m respectively to emulate slow scanning worms. The objective of the experiment was to demonstrate the behavior of the proposed detection scheme in detecting slow scanning malicious worm attacks on a target network.

Fig.8(a), Fig.8(b) and Fig.8(c) show snapshots of results from both the worm correlation algorithm (WCA) and slow worm correlation algorithm (SWCA). Fig.8(a) and Fig.8(b) show that though the WCE received both normal scanning worm profiles and slow worm traffic profiles during the worm detection window, the combined detection probability, P_{Di} and false detection probability, P_{Fi} for the normal worm traffic profiles met the criteria for detection ($P_{Di} > \Phi_D = 0.8$, $P_{Fi} < \Phi_F = 0.2$) by the WCA.

Also, the window-based profiler ensures that only the slow worm profiles with propagation rates we are interested in are forwarded to the SWCE. On the SWCE, Fig.8(c) shows that the likelihood ratio computed for the slow worm traffic profiles also met criteria in (10) for detection and therefore were selected by the SWCA.

[7] Note that our emulation of host-based detection with snort-based alerts and real-time logging was only used to demonstrate the behavior of the proposed detection technique. Other host-based AIDS software such as Thirdbrigade host AIDS, Cisco Security Agent and Tripwire host AIDS can be used for detection in enterprise deployments.

V. Conclusion and Future Work

In this paper, we proposed a detection technique for slow worms based on the Generalized Evidence Processing (GEP) theory, a sensor integration and data fusion technique. With GEP theory, evidence collected by distributed detectors determine the probability associated with a detection decision under a hypothesis. The evidence are combined at a fusion center (a gateway router) to arrive at an optimal fused detection decision by minimizing a cummulative decision risk function.

We emphasized that slow worms do not exist alone in the wild. Typically, malicious traffic flows of varying scanning rates can occur in the wild, and detection of slow scanning worms in particular can be challenging in such a scenario due to interference from faster scanning traffic flows. We therefore used a detection window-based self adapting profiler to filter detected malicious traffic profiles with scanning rates greater than the low scanning rates we are interested in. We experimented with the proposed detection scheme on a live test-bed to demonstrate the behavior of the detection technique.

For future work, we intend to experiment with more complex network and traffic scenarios. We also intend to investigate the impact of indecisive detectors on GEP based intrusion detection of malicious worms.

References

[1] D. Dash, B. Kveton, J. Agosta, E. Schooler, J. Chandrashekar, A. Bachrach, and A. Newman, "When gossip is good: Distributed probabilistic inference for detection of slow network intrusions," in *21st National Conference on Artificial Intelligence (AAAI06), 2006*, 2006.

[2] J. Li, S. Stafford, and T. Ehrenkranz, "On the performance of SWORD in detecting zero-day-worm-infected hosts," in *Summer Simulation Multiconference (SummerSim 2006)*, 2006.

[3] V. Sekar, Y. Xie, M. Reiter, and H. Zhang, "A multi-resolution approach for worm detection and containment," in *DSN '06: Proceedings of the International Conference on Dependable Systems and Networks*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 189–198.

[4] M. Costa, J. Crowcroft, M. Castro, A. Rowstron, L. Zhou, L. Zhang, and P. Barham, "Vigilante: End-to-end containment of Internet worms," in *Proceedings of the Symposium on Systems and Operating Systems Principles (SOSP)*, 2005, pp. 133–147.
[5] C. Gates and C. Taylor, "Challenging the anomaly detection paradigm: A provocative discussion," in *NSPW '06: Proceedings of the 2006 workshop on New security paradigms*. New York, NY, USA: ACM, 2006, pp. 21–29.
[6] D. Mutz, F. Valeur, C. Kruegel, and G. Vigna, "Anomalous system call detection," *ACM Transactions on Information and System Security*, vol. 9, pp. 61–93, 2006.
[7] C. Sullivan, *Cisco Security Agent*. Cisco Press, 2005.
[8] S. Singh, C. Estan, G. Varghese, and S. Savage, "Automated worm fingerprinting," in *Operating Systems Design and Implementation (OSDI), Proceedings of the 6th conference on Symposium on Operating Systems Design Implementation - Volume 6*, 2004, pp. 45–60.
[9] S. Thomopoulos, "Sensor integration and data fusion," *Journal of Robotic Systems*, vol. 7, no. 3, pp. 337–372, 1990.
[10] ——, "Theories in distributed data fusion: Comparison and generalization," *SPIE*, vol. 1383, p. 623, 1990.
[11] T. M.C. and V. Venkataramanan, "Dempster-shafer theory for intrusion detection in ad hoc networks," *IEEE Internet Computing*, vol. 9, no. 6, pp. 35–41, 2005.
[12] C. Siaterlis and B. Maglaris, "Towards multisensor data fusion for DoS detection," in *Proceedings of the 2004 ACM symposium on Applied computing*. ACM Press, 2004, pp. 439–446.
[13] L. Klein, *Sensor and Data Fusion: A Tool for Information Assessment and Decision Making (SPIE Press Monograph Vol. PM138)*. SPIE - International Society for Optical Engineering, 2004.
[14] D. Hall and S. McMullen, *Mathematical Techniques in Multisensor Data Fusion*. Norwood, MA, USA: Artech House, Inc., 2004.
[15] F. Akujobi, I. Lambadaris, and E. Kranakis, "An Integrated Approach to Detection of Fast and Slow Scanning Worms," *ACM Symposium on Information, Computer and Communications Security (ASIACCS 2009)*, 2009.
[16] Network Security Resources, "Governmentsecurity.org," http://www.governmentsecurity.org/forum/index.php?showtopic=4726, 2003. Website was functional in 2006.
[17] C. Shannon and D. Moore, "The spread of the witty worm," in *IEEE Security Privacy, vol. 2, no. 4*, 2004.
[18] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver, "Inside the slammer worm," in *IEEE Security Privacy, vol. 1, no. 4*, 2003.
[19] Swsoft, "Openvz homepage," http://openvz.org/, 2008.

APPENDIX I
GENERALIZED EVIDENCE PROCESSING THEORY

In this section, following the GEP theory [9], we determine the decision rules which ensure the fused decision made at the fusion center minimizes the cummulative decision risk $\mathbf{R}$. The cummulative risk $\mathbf{R}$ can be expressed as:

$$\mathbf{R} = \sum_a \sum_b C_{ab} P(H_b) \int_{\mathbf{D}=a} dP(\underline{d}|H_b); b = 0, 1 \text{ and } a = 0, 1, 2 \quad (11)$$

where $\mathbf{D} = 0, 1, 2$ are the fused decisions that "H_0 is true", "H_1 is true" and "H_0 or H_1 is true" respectively which occupy the fused decision space. Solving,

$$\begin{aligned}\mathbf{R} = &\int_{\mathbf{D}=0} [P(H_0)C_{00}dP(\underline{d}|H_0) + P(H_1)C_{01}dP(\underline{d}|H_1)] \\ &+ \int_{\mathbf{D}=1} [P(H_0)C_{10}dP(\underline{d}|H_0) + P(H_1)C_{11}dP(\underline{d}|H_1)] \\ &+ \int_{\mathbf{D}=2} [P(H_0)C_{20}dP(\underline{d}|H_0) + P(H_1)C_{21}dP(\underline{d}|H_1)]\end{aligned}$$

$\mathbf{R}$ is minimized if the fusion decision rule assigns $\mathbf{D}$ (the fused decision) to the region ($\mathbf{D} = 0$, $\mathbf{D} = 1$ or $\mathbf{D} = 2$) that results in the least integrand under the three integrals. Since $\underline{d}$ is a vector with discrete components, we can write the fusion decision rules as follows:

$$P_0 C_{00} P(\underline{d}|H_0) + P_1 C_{01} P(\underline{d}|H_1) \underset{\mathbf{D}=0 \text{ or } \mathbf{D}=2}{\overset{\mathbf{D}=1 \text{ or } \mathbf{D}=2}{\gtrless}} P_0 C_{10} P(\underline{d}|H_0) + P_1 C_{11} P(\underline{d}|H_1)$$

$$P_0 C_{00} P(\underline{d}|H_0) + P_1 C_{01} P(\underline{d}|H_1) \underset{\mathbf{D}=0 \text{ or } \mathbf{D}=1}{\overset{\mathbf{D}=2 \text{ or } \mathbf{D}=1}{\gtrless}} P_0 C_{20} P(\underline{d}|H_0) + P_1 C_{21} P(\underline{d}|H_1)$$

$$P_0 C_{20} P(\underline{d}|H_0) + P_1 C_{21} P(\underline{d}|H_1) \underset{\mathbf{D}=2 \text{ or } \mathbf{D}=0}{\overset{\mathbf{D}=1 \text{ or } \mathbf{D}=0}{\gtrless}} P_0 C_{10} P(\underline{d}|H_0) + P_1 C_{11} P(\underline{d}|H_1)$$

where,

$$P_1 = P(H_1) \quad \text{and} \quad P_0 = P(H_0)$$

Dividing both sides by $P(\underline{d}|H_0)$ and defining $\Lambda(\underline{d}) = \frac{P(\underline{d}|H_1)}{P(\underline{d}|H_0)}$ the decision rules become:

$$[C_{01} - C_{11}]\Lambda(\underline{d}) \underset{\mathbf{D}=0 \text{ or } \mathbf{D}=2}{\overset{\mathbf{D}=1 \text{ or } \mathbf{D}=2}{\gtrless}} \frac{P(H_0)}{P(H_1)}[C_{10} - C_{00}]$$

$$[C_{01} - C_{21}]\Lambda(\underline{d}) \underset{\mathbf{D}=0 \text{ or } \mathbf{D}=1}{\overset{\mathbf{D}=2 \text{ or } \mathbf{D}=1}{\gtrless}} \frac{P(H_0)}{P(H_1)}[C_{20} - C_{00}]$$

$$[C_{21} - C_{11}]\Lambda(\underline{d}) \underset{\mathbf{D}=2 \text{ or } \mathbf{D}=0}{\overset{\mathbf{D}=1 \text{ or } \mathbf{D}=0}{\gtrless}} \frac{P(H_0)}{P(H_1)}[C_{10} - C_{20}]$$

We assume that there is no penalty for a correct decision, hence the associated cost for a correct decision is zero (i.e. $C_{00} = C_{11} = 0$). Solving,

$$\Lambda(\underline{d}) \underset{\mathbf{D}=0 \text{ or } \mathbf{D}=2}{\overset{\mathbf{D}=1 \text{ or } \mathbf{D}=2}{\gtrless}} \frac{P(H_0)}{P(H_1)} \frac{C_{10}}{C_{01}} \quad (12)$$

$$\Lambda(\underline{d}) \underset{\mathbf{D}=0 \text{ or } \mathbf{D}=1}{\overset{\mathbf{D}=2 \text{ or } \mathbf{D}=1}{\gtrless}} \frac{P(H_0)}{P(H_1)} \frac{C_{20}}{C_{01} - C_{21}} \quad (13)$$

$$\Lambda(\underline{d}) \underset{\mathbf{D}=2 \text{ or } \mathbf{D}=0}{\overset{\mathbf{D}=1 \text{ or } \mathbf{D}=0}{\gtrless}} \frac{P(H_0)}{P(H_1)} \frac{C_{10} - C_{20}}{C_{21}} \quad (14)$$