Effective Acquaintance Management based on Bayesian Learning for Distributed Intrusion Detection Networks

Carol J Fung, *Member, IEEE,* Jie Zhang, *Member, IEEE,*
Raouf Boutaba, *Senior Member, IEEE,*

***Abstract*—An effective Collaborative Intrusion Detection Network (CIDN) allows distributed Intrusion Detection Systems (IDSes) to collaborate and share their knowledge and opinions about intrusions, to enhance the overall accuracy of intrusion assessment as well as the ability of detecting new classes of intrusions. Toward this goal, we propose a distributed Host-based IDS (HIDS) collaboration system, particularly focusing on acquaintance management where each HIDS selects and maintains a list of collaborators from which they can consult about intrusions. Specifically, each HIDS evaluates both the false positive (FP) rate and false negative (FN) rate of its neighboring HIDSes' opinions about intrusions using Bayesian learning, and aggregates these opinions using a Bayesian decision model. Our dynamic acquaintance management algorithm allows each HIDS to effectively select a set of collaborators. We evaluate our system based on a simulated collaborative HIDS network. The experimental results demonstrate the convergence, stability, robustness, and incentive-compatibility of our system.**

Keywords: Host-based Intrusion Detection Systems, Acquaintance Management, Collaborative Networks, Computer Security.

I. Introduction

In recent years, cyber attacks from the Internet are becoming more sophisticated and harder to detect. Intrusions can have many forms such as worms, spamware, viruses, spyware, denial-of-service attacks (DoS), malicious logins, etc. The potential damage of these intrusions can be significant if they are not detected promptly. A recent example is the Conficker worm which infected more than 3 million Microsoft server systems during the year of 2008 to 2009, with the estimated economic loss of 9.1 billion dollars [6]. Contemporary intrusion attacks compromise a large number of nodes to form botnets. Attackers not only harvest private data and identify information from compromised nodes, but also use those compromised nodes to launch attacks such as distributed-denial-of-service (DDoS) attacks [25], distribution of spam messages, or organized phishing attacks [18].

To protect computer users from malicious intrusions, *Intrusion Detection Systems (IDSes)* are designed to monitor network traffic and computer activities and to raise intrusion alerts to network administrators or security officers. IDSes can be categorized into host-based Intrusion Detection Systems (HIDSes) or network-based Intrusion Detection Systems (NIDSes) according to their targets, and into signature-based IDS or anomaly-based IDS according to their detection methodologies.

A NIDS monitors the network traffic from/to one or a group of computers and compare the data with known intrusion patterns. Examples of NIDS are Snort [4] and Bro [1]. A HIDS identify intrusions by comparing observable intrusion data such as log files and computer activities against suspicious patterns. Examples of HIDSes are OSSEC [2], an anti-virus software, and Tripwire [5]. A signature-based IDS identifies malicious code if a match is found with a pattern in the attack signature database. An anomaly-based IDS [21], [29], on the other hand, monitors the traffic or behavior of the computer and classifies that as either normal or anomalous based on some heuristics or rules, rather than patterns or signatures. Alerts are raised when anomalous activities are found. Compared to NIDS, a HIDS has a deeper sight into each host so it can detect intrusions which are hard to identify by observing network traffic only. However, a single HIDS lacks an overall view of the network and can be easily compromised by new attacks. Collaboration among HIDSes can effectively improve the overall intrusion detection efficiency by using collective information from collaborative HIDSes. In this paper, we focus on the collaboration among HIDSes. A signature-based IDS can accurately identify intrusions and the false positive rate is low compared to anomaly-based detection. However, it is not effective for zero-day attacks, polymorphic, and metamorphic malware [22]. An anomaly-based IDS may detect zero-day attacks by analyzing their abnormal behaviors. However, an anomaly-based detection usually generates a high false positive rate.

Traditional HIDSes work in isolation and they rely on downloading signatures or rules from their vendors to detect intrusions. They may be easily compromised by unknown or new threats since not a single vendor has comprehensive knowledge about all intrusions. Collaboration among HIDSes can overcome this weakness by having each peer HIDS benefit from the collective knowledge and experience shared by other peers. This enhances the overall accuracy of intrusion assessment as well as the ability to detect new classes of intrusions. A *Collaborative Intrusion Detection Network* (CIDN) is an overlay network which provides a collaboration infrastructure

This paper extends our previous work in [16] with a detailed description of the design of our CIDN framework as well as extended evaluation for the robustness of our acquaintance management and its comparison with other acquaintance management algorithms. The related work section has also been greatly extended to include detailed discussions on the state of the art in the area of collaborative intrusion detection, the work on trust modeling in collaborative intrusion detection systems, as well as the collaborative intrusion decision models in the literature.

for HIDSes to share information and experience with each other to achieve improved detection accuracy. The topology of a CIDN can be centralized, such as DShield [31], CRIM [8], and N-version AV [26], or distributed, such as Indra [19], NetShield [7], and Host-based CIDN [13]. Note that the CIDN collaboration framework is also applicable to NIDSes.

However, in a CIDN, malicious insiders may send false information to mislead other HIDSes to make incorrect intrusion decisions, and in this way, render the collaboration system not useful. Furthermore, HIDSes in the collaboration network may have different intrusion detection expertise levels and capabilities. How a HIDS selects collaborators to achieve optimal efficiency is an important problem to solve for a CIDN. We define a CIDN *acquaintance management* as the process of identifying, selecting, and maintaining collaborators for each HIDS. An effective acquaintance management is crucial to the design of a CIDN.

We provide a Bayesian learning technique that helps each HIDS identify expert nodes and novice nodes based on past experience with them. Specifically, the false positive (FP) rate and false negative (FN) rate of each collaborator. Dishonest collaborators are identified and removed from its collaborator list. We define *feedback aggregation* in CIDN as a decision method whether or not to raise an alarm based on the collected opinions (feedback) from collaborator HIDSes. We propose a Bayesian decision model for feedback aggregation. Bayes theory is used to estimate the conditional probability of intrusions based on feedback from collaborators. A cost function is modeled to include the false positive decision cost and false negative decision cost. A decision of whether to raise alarm or not is chosen to achieve the minimal cost of false decisions.

For collaborator selection, a HIDS may add all honest HIDSes into its collaborator list to achieve maximized detection accuracy. However, including a large list of collaborators may result in high maintenance cost. We define an *acquaintance selection* as the process to find the list of collaborators to minimize false decision cost and maintenance cost. Existing approaches for acquaintance management often set a fixed number of collaborators [34], or a fixed accuracy threshold to filter out less honest or low expertise collaborators [35], [14], [15]. These static approaches lack flexibility, and the fixed acquaintance length or accuracy threshold may not be optimal when the context changes (e.g. some nodes leave the network and some new nodes join the network). Our proposed acquaintance management algorithm can dynamically select collaborators in any context setting to obtain high efficiency at minimum cost.

For collaborator maintenance, the HIDSes in our system periodically update their collaborators lists to guarantee an optimal cost. A probation list is used to explore and learn the quality of new potential collaborators. New collaborators stay in the probation list for a certain period before their feedback is considered for intrusion decision.

We evaluate our system using a simulated collaboration network using a Java-based discrete-event simulation framework. The results show that the proposed Bayesian decision model outperforms the threshold-based model [26], which only counts the number of intrusion reports, in terms of false decision cost. The results also show that our dynamic acquaintance management algorithm outperforms the static approaches of setting a fixed acquaintance length or accuracy threshold. Finally, our approach also achieves several desired properties, such as efficiency, stability, robustness, and incentive-compatibility.

Major contributions of our work are summarized as follows:

1) A novel Bayesian decision model is proposed for feedback aggregation in collaborative intrusion detection systems to achieve minimal false decision cost;
2) An acquaintance selection algorithm is devised to optimally select collaborators, which leads to minimal overall cost including false decision cost and maintenance cost;
3) A dynamic acquaintance management algorithm is proposed to integrate the concept of probation period and consensus negotiation;

The rest of the paper is organized as follows. Section II gives a brief introduction of CIDNs and some of their important components. Section III describes our formalization of HIDS learning model and feedback aggregation. Acquaintance selection and management algorithms are presented in Section IV. We discuss several potential threats to our system and corresponding defenses in Section V. We then present evaluation results demonstrating the effectiveness of our acquaintance management and its desired properties in Section VI. We discuss some related work in Section VII, and conclude this paper in Section VIII.

II. Collaborative Intrusion Detection Network

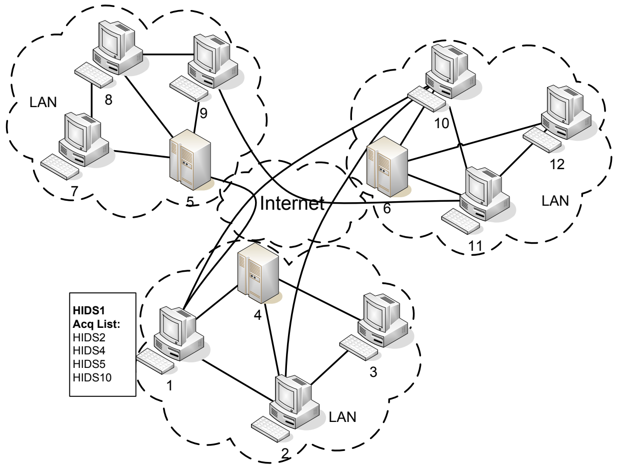

Fig. 1. Overlay Network of Collaborating HIDSes

A Collaborative Intrusion Detection Network (CIDN) is shown in Figure 1 as an overlay network of collaborating HIDSes. HIDSes from different vendors are connected in a peer-to-peer manner. Each peer HIDS maintains a list of collaborators. We use the terminology *Acquaintance List* to represent the list of collaborators for each HIDS and use the terms collaborator and acquaintance interchangeably in this paper. Note that the relationship of collaboration is symmetric. i.e., if HIDS X is in HIDS Y's acquaintance list, then HIDS

Y is also in HIDS X's acquaintance list. HIDSes may have different expertise levels in intrusion detection. They can be compromised or dishonest when providing feedback to collaborators. They may also be self-interest driven when providing assistance to other peers.

When a HIDS detects suspicious behavior for example from an executable file but lacks confidence to make a decision whether it should raise an alarm or not, it may send a *consultation request* to its collaborators for diagnosis. The consultation request contains the description of the suspicious activities or the executable file, including the suspicious in/out communication traffic, the log entries of suspicious activities, or the binaries source of the suspicious activities. The feedback from collaborators contains the assessment result of the request, which is either a "yes" for under-attack or a "no" for no-attack. All feedback will be aggregated and a final decision is made based on the aggregation result. However, a malicious (or malfunctioning) HIDS in a CIDN may send false intrusion assessments to its collaborators, resulting in false positive or false negative decisions made by the requesters. It is thus important to evaluate peer HIDSes and choose the ones that likely provide higher detection accuracy. Acquaintance learning allows a HIDS to evaluate the detection accuracy of its collaborators based on its past experience with them.

In our system, HIDSes use *test messages* to evaluate the detection accuracy and truthfulness of other HIDSes. Test messages are "bogus" consultation requests, and sent out in a way that makes them difficult to be distinguished from real consultation requests. The testing node needs to know beforehand the true diagnosis result of the test message and compare it with the received feedback to derive detection accuracy of other HIDSes. For example, the samples of known malware and legitimate files can be used as test messages. The usage of test messages helps with identifying inexperienced and/or malicious nodes within the network. The idea of "test messages" was previously introduced in [30] and [13]. It is adopted in our CIDN for each HIDS to gain experience with others quickly at affordable communication cost, and to evaluate the detection accuracy of its acquaintances.

Collaborative intrusion decision is a process for a HIDS to aggregate the intrusion assessments (feedback) from its collaborators and make an overall decision. The evaluation result of acquaintances' detection accuracy is used as a parameter in the feedback aggregation function to improve collaborative intrusion detection accuracy. The acquaintance management system updates acquaintance list periodically to recruit new collaborators and/or remove unwanted ones. The collaboration relationship is established based on mutual consensus, i.e., it is established only if both sides agree. We will elaborate on our acquaintance management for CIDN in Section IV.

III. HIDS Detection Accuracy Evaluation and Feedback Aggregation

To select collaborators, a HIDS should first learn the qualification of all candidate HIDSes. In this section, we first introduce a Bayesian learning model to evaluate the detection accuracy of the candidates. A Bayesian decision model is then used to optimally aggregate feedback from acquaintances.

TABLE I
Summary of Notations

Symbol	Meaning
$X \in \{0,1\}$	Random variable denoting whether there is an attack or not
$Y \in \{0,1\}$	Random variable of positive or negative diagnose from a HIDS
$\mathbf{y}$	A feedback instance vector from acquaintances
$\mathbf{Y}$	Feedback vector from acquaintances
$\mathcal{C}$	Set of acquaintance candidates
$\mathcal{A}$	Set of Acquaintances
l	The acquaintance list length
δ	The decision of raising alarm or not
$R(.)$	The risk cost of false alarms and miss intrusions
$M(.)$	The maintenance cost of acquaintances
C_{fp}, C_{fn}	Unit cost of false alarm and miss intrusion
C_a	Unit cost of maintaining each acquaintance
π_0, π_1	Priory probability of no-intrusion and with-intrusion
T_i, F_i	True positive rate and false positive rate of IDS i
λ	Forgetting factor of the past experience

A. Detection Accuracy for a Single HIDS

To better capture the qualification of a HIDS, we use both *false positive* (FP) and *true positive* (TP) rates to represent the detection accuracy of a HIDS. Let $\mathcal{A}$ denote the set of acquaintances and random variables F_k and T_k denote the FP and TP rates of acquaintance $k \in \mathcal{A}$ respectively. FP is the probability that the HIDS gives a positive diagnosis (under-attack) under the condition of no-attack, and TP is the probability that the HIDS gives a correct positive diagnosis under the condition of under-attack. Let random variable $X \in \{0,1\}$ represent the random event on whether there is an attack or not, and let random variable $Y \in \{0,1\}$ denote whether the HIDS makes a positive diagnosis or not. Then FP and TP can be written as $\mathbb{P}[Y=1|X=0]$ and $\mathbb{P}[Y=1|X=1]$, respectively. The list of notations is summarized in Table I.

Let $\mathcal{F}_k$ and $\mathcal{T}_k$ be the probability density functions of F_k and T_k whose support is $[0,1]$. We use the notation $Z_0 : Y_k = 1|X=0$ and $Z_1 : Y_k = 1|X=1$ to represent the conditional variables that acquaintance k gives positive decision under the conditions where there is no attack and there is an attack respectively. They can be seen as two independent random variables satisfying Bernoulli distribution with successful rates F_k and T_k, respectively. The past experience with acquaintance k can be seen as the samples from the Bernoulli distributions. According to the Bayesian probability theory [17], the posterior distribution of $\mathcal{F}_k$ and $\mathcal{T}_k$ given a set of observed samples can be represented using a Beta function, written as follows:

$$\mathcal{F}_k \sim \quad \text{Beta}(x_k|\alpha_k^0, \beta_k^0) = \frac{\Gamma(\alpha_k^0+\beta_k^0)}{\Gamma(\alpha_k^0)\Gamma(\beta_i^0)} x_k^{\alpha_k^0-1}(1-x_k)^{\beta_k^0-1}, \quad (1)$$

$$\mathcal{T}_k \sim \quad \text{Beta}(y_k|\alpha_k^1, \beta_k^1) = \frac{\Gamma(\alpha_k^1+\beta_k^1)}{\Gamma(\alpha_k^1)\Gamma(\beta_i^1)} y_k^{\alpha_k^1-1}(1-y_k)^{\beta_k^1-1}, \quad (2)$$

where $\Gamma(\cdot)$ is the gamma function [20], and its parameters α_k^0, α_k^1 and β_k^0, β_k^1 are given by

$$\alpha_k^0 = \sum_{j=1}^{u} \lambda^{t_{k,j}^0} r_{k,j}^0 \qquad \beta_k^0 = \sum_{j=1}^{u} \lambda^{t_{k,j}^0}(1-r_{k,j}^0);$$
$$\alpha_k^1 = \sum_{j=1}^{v} \lambda^{t_{k,j}^1} r_{k,j}^1 \qquad \beta_k^1 = \sum_{j=1}^{v} \lambda^{t_{k,j}^1}(1-r_{k,j}^1), \quad (3)$$

where $\alpha_k^0, \beta_k^0, \alpha_k^1, \beta_k^1$ are the cumulated instances of false positive, true negative, true positive, and false negative, respectively, from acquaintance k. $r_{k,j}^0 \in \{0,1\}$ is the jth diagnosis result from acquaintance k under no-attack. $r_{k,j}^0 = 1$ means the diagnosis from k is positive while there is actually no attack happening. $r_{k,j}^0 = 0$ means otherwise. Similarly, $r_{k,j}^1 \in \{0,1\}$ is the jth diagnosis data from acquaintance k under attack where $r_{k,0}^1 = 1$ means that the diagnosis from k is positive under attack, and $r_{k,0}^1 = 0$ means otherwise. Parameters $t_{k,j}^0$ and $t_{k,j}^1$ denote the time elapsed since the jth feedback is received. $\lambda \in [0,1]$ is the forgetting factor on the past experience. A small λ makes old observations quickly forgettable. We use exponential moving average to accumulate past experience so that old experience takes less weight than new experience. u is the total number of no-attack cases among the past records and v is the total number of attack cases.

To make the parametric updates scalable to data storage and memory, we can use the following recursive formula to update α_k^0, α_k^1 and β_k^0, β_k^1:

$$\begin{aligned} \alpha_k^m(t_j) &= \lambda^{(t_{k,j}^m - t_{k,j-1}^m)} \alpha_k^m(t_{k,j-1}^m) + r_{k,j}^m; \\ \beta_k^m(t_j) &= \lambda^{(t_{k,j}^m - t_{k,j-1}^m)} \beta_k^m(t_{k,j-1}^m) + r_{k,j}^m, \end{aligned} \quad (4)$$

where $l = 0,1$ and $j-1$ indexes the previous data point used for updating α_k^m or β_k^m. Through this way, only the previous state and the current state are required to be recorded, which is efficient in terms of storage compared to when all states are recorded in Equation 3.

B. Feedback Aggregation

When a HIDS detects suspicious activities and is not confident about its decision, it sends out the description of the suspicious activities or the related executable files to its collaborators for consultation. The node receives diagnosis results from its collaborators, denoted by vector $\mathbf{y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_{|\mathcal{A}|}\}$, where $\mathbf{y}_i \in \{0,1\}$, for $0 < i < |\mathcal{A}|$, is the feedback from acquaintance i. We use $X \in \{0,1\}$ to denote the scenario of "no-attack" or "under-attack", and $\mathbf{Y} \in \{0,1\}^{|\mathcal{A}|}$ to denote all possible feedback from acquaintances. The conditional probability of a HIDS being "under-attack" given the diagnosis results from all acquaintances can be written as $\mathbb{P}[X=1|\mathbf{Y}=\mathbf{y}]$. Using Bayes' Theorem [28] and assuming that the acquaintances provide diagnoses independently and their FP rate and TP rate are known, we have

$$\begin{aligned} \mathbb{P}[X{=}1|\mathbf{Y}{=}\mathbf{y}] &= \frac{\mathbb{P}[\mathbf{Y}{=}\mathbf{y}|X{=}1]\mathbb{P}[X{=}1]}{\mathbb{P}[\mathbf{Y}{=}\mathbf{y}|X{=}1]\mathbb{P}[X{=}1]+\mathbb{P}[\mathbf{Y}{=}\mathbf{y}|X{=}0]\mathbb{P}[X{=}0]} \\ &= \frac{\pi_1 \prod_{k=1}^{|\mathcal{A}|} T_k^{\mathbf{y}_k}(1-T_k)^{1-\mathbf{y}_k}}{\pi_1 \prod_{k=1}^{|\mathcal{A}|} T_k^{\mathbf{y}_k}(1-T_k)^{1-\mathbf{y}_k} + \pi_0 \prod_{k=1}^{|\mathcal{A}|} F_k^{\mathbf{y}_k}(1-F_k)^{1-\mathbf{y}_k}}, \end{aligned}$$

where $\pi_0 = \mathbb{P}[X=0]$ and $\pi_1 = \mathbb{P}[X=1]$, such that $\pi_0 + \pi_1 = 1$, are the prior probabilities of the scenarios of "no-attack" and "under-attack", respectively. $\mathbf{y}_k \in \{0,1\}$ is the kth element of vector $\mathbf{y}$.

Since T_k and F_k are both random variables with distributions as in Equations (1) and (2), we can see that the conditional probability $\mathbb{P}[X=1|\mathbf{Y}=\mathbf{y}]$ is also a random variable. We use a random variable P to denote $\mathbb{P}[X=1|\mathbf{Y}=\mathbf{y}]$. Then P takes a continuous value over domain $[0,1]$. We use $f_P(p)$ to denote the probability density function of P.

When α and β are sufficiently large, a Beta distribution can be approximated by Gaussian distribution according to $\text{Beta}(\alpha,\beta) \approx N\left(\frac{\alpha}{\alpha+\beta}, \sqrt{\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}}\right)$. Then the density function of P can be also approximated using Gaussian distribution. By Gauss's approximation formula, we have,

$$\begin{aligned} \mathbb{E}[P] &\approx \frac{1}{1+\frac{\pi_0 \prod_{k=1}^{|\mathcal{A}|} \mathbb{E}[F_k]^{\mathbf{y}_k}(1-\mathbb{E}[F_k])^{1-\mathbf{y}_k}}{\pi_1 \prod_{k=1}^{|\mathcal{A}|} \mathbb{E}[T_k]^{\mathbf{y}_k}(1-\mathbb{E}[T_k])^{1-\mathbf{y}_k}}} \\ &= \frac{1}{1+\frac{\pi_0}{\pi_1}\prod_{k=1}^{|\mathcal{A}|} \frac{\alpha_k^1+\beta_k^1}{\alpha_k^0+\beta_k^0}(\frac{\alpha_k^0}{\alpha_k^1})^{\mathbf{y}_k}(\frac{\beta_k^0}{\beta_k^1})^{1-\mathbf{y}_k}}. \end{aligned} \quad (5)$$

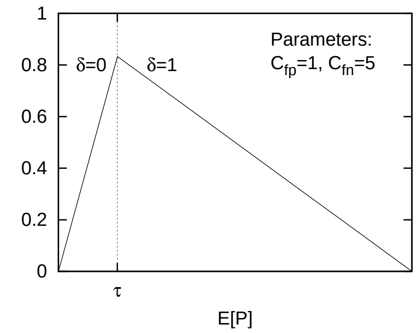

Fig. 2. Bayes Risk for Optimal Decisions when $C_{fp} = 1$ and $C_{fn} = 5$

Let C_{fp} and C_{fn} denote the marginal cost of a FP decision and a FN decision. We assume there is no cost when a correct decision is made. We use marginal cost because the cost of a FP may change in time depending on the current state. C_{fn} largely depends on the potential damage level of the attack. For example, an intruder intending to track a user's browsing history may have lower C_{fn} than an intruder intending to modify a system file. We define a decision function $\delta(\mathbf{y}) \in \{0,1\}$, where $\delta = 1$ means raising an alarm and $\delta = 0$ means no alarm. Then, the Bayes risk can be written as:

$$\begin{aligned} R(\delta) &= \int_0^1 (C_{fp}(1-x)\delta + C_{fn}x(1-\delta))f_P(x)dx \\ &= \delta C_{fp}\int_0^1 (1-p)f_P(p)dp + (1-\delta)C_{fn}\int_0^1 pf_P(p)dp \\ &= \int_0^1 C_{fn}xf_P(x)dx + \delta\left(C_{fp} - (C_{fp}+C_{fn})\int_0^1 xf_P(x)dx\right) \\ &= C_{fn}\mathbb{E}[P] + \delta(C_{fp} - (C_{fp}+C_{fn})\mathbb{E}[P]), \end{aligned} \quad (6)$$

where $f_P(p)$ is the density function of P. To minimize the risk $R(\delta)$, we need to minimize $\delta(C_{fp} - (C_{fp}+C_{fn})\mathbb{E}[P])$. Therefore, we raise an alarm (i.e. $\delta = 1$) if

$$\mathbb{E}[P] \geq \frac{C_{fp}}{C_{fp}+C_{fn}}. \quad (7)$$

Let $\tau = \frac{C_{fp}}{C_{fp}+C_{fn}}$ be the threshold. If $\mathbb{E}[P] \geq \tau$, we raise an alarm, otherwise no alarm is raised. The corresponding Bayes risk for the optimal decision is:

$$R(\delta) = \begin{cases} C_{fp}(1-\mathbb{E}[P]) & \text{if } \mathbb{E}[P] \geq \tau, \\ \\ C_{fn}\mathbb{E}[P] & \text{otherwise.} \end{cases} \tag{8}$$

An example of the Bayes risk for optimal decisions when $C_{fp} = 1$ and $C_{fn} = 5$ is illustrated in Figure 2.

IV. Acquaintance Management

Intuitively when a HIDS consults a larger number of acquaintances, it can achieve higher detection accuracy and lower risk of being compromised. However, having more acquaintances causes higher maintenance cost since the HIDS needs to allocate resources for each node in its acquaintance list. When a HIDS makes a decision about how many acquaintances to recruit, both the intrusion risk cost and the maintenance cost should be taken into account. When adding a node as an acquaintance does not lower the total cost, then the node shall not be added into the acquaintance list. However, how to select acquaintances and how many acquaintances to include are crucial to build an efficient CIDN. In this section, we first define the acquaintance selection problem, then a corresponding solution is devised to find the optimal set of acquaintances. Finally, we propose an acquaintance management algorithm for HIDSes to learn, recruit, update, or remove their acquaintances dynamically.

A. Problem Statement

Let $\mathcal{A}_i$ denote the set of acquaintances of HIDS i. Let $M_i(\mathcal{A}_i)$ be the cost for HIDS i to maintain the acquaintance set $\mathcal{A}_i$. We use $R_i(\mathcal{A}_i)$ to denote the risk cost of missing intrusions and/or false alarms for HIDS i, given the feedback of acquaintance set $\mathcal{A}_i$. In the rest of this section, we drop subscript i from our notations for the convenience of presentation.

Our goal is to select a set of acquaintances from a list of candidates so that the overall cost $R(\mathcal{A})+M(\mathcal{A})$ is minimized. We define the problem as follows:

Given a list of acquaintance candidates $\mathcal{C}$, we need to find a subset of acquaintances $\mathcal{A} \subseteq \mathcal{C}$, such that the overall cost $R(\mathcal{A}) + M(\mathcal{A})$ is minimized.

In practice, maintenance cost of acquaintances may not be negligible since acquaintances send test messages/consultations periodically to ask for diagnosis. It takes resources (CPU and memory) for the HIDS to receive, analyze the requests, and reply with corresponding answers. The selection of $M_i(.)$ can be user defined on each host. For example, a simple maximum acquaintance length restriction can be mapped to $M(\mathcal{A}) = C\max(|\mathcal{A}| - L, 0)$, where $L \in \mathcal{N}^+$ is the acquaintance length upper-bound and $C \in [0, \infty)$ is the penalty of exceeding the bound.

The risk cost can be expressed as:

$$\begin{aligned} R(\mathcal{A}) = {} & C_{fn}P[\delta = 0|X = 1]P[X = 1] \\ & + C_{fp}P[\delta = 1|X = 0]P[X = 0] \end{aligned}$$

where C_{fn}, C_{fp} denote the marginal cost of missing an intrusion and raising a false alarm, respectively. $P[X = 1] = \pi_1, P[X = 0] = \pi_0$ are the prior probabilities of under-attack and no-attack, where $\pi_0 + \pi_1 = 1$. Note that in practice π_1 can be learned from the history and be updated whenever a new threat is found. A moving average method can be used to update the estimated value.

The above equation can be further written as:

$$\begin{aligned} R(\mathcal{A}) = {} & C_{fn}\pi_1 \sum_{\forall \mathbf{y}\in\{0,1\}^{|\mathcal{A}|}|\delta(\mathbf{y})=0} P[\mathbf{Y} = \mathbf{y}|X = 1] \\ & + C_{fp}\pi_0 \sum_{\forall \mathbf{y}\in\{0,1\}^{|\mathcal{A}|}|\delta(\mathbf{y})=1} P[\mathbf{Y} = \mathbf{y}|X = 0] \\ = {} & C_{fn}\pi_1 \sum_{\forall \mathbf{y}\in\{0,1\}^{|\mathcal{A}|}|\delta(\mathbf{y})=0} \prod_{i=1}^{|\mathcal{A}|} (T_i)^{\mathbf{y}_i}(1-T_i)^{1-\mathbf{y}_i} \\ & + C_{fp}\pi_0 \sum_{\forall \mathbf{y}\in\{0,1\}^{|\mathcal{A}|}|\delta(\mathbf{y})=1} \prod_{i=1}^{|\mathcal{A}|} (F_i)^{\mathbf{y}_i}(1-F_i)^{1-\mathbf{y}_i} \\ = {} & C_{fn}\pi_1 \sum_{\forall \mathbf{y}\in\{0,1\}^{|\mathcal{A}|}|f(\mathbf{y})<1} \prod_{i=1}^{|\mathcal{A}|} (T_i)^{\mathbf{y}_i}(1-T_i)^{1-\mathbf{y}_i} \\ & + C_{fp}\pi_0 \sum_{\forall \mathbf{y}\in\{0,1\}^{|\mathcal{A}|}|f(\mathbf{y})\geq 1} \prod_{i=1}^{|\mathcal{A}|} (F_i)^{\mathbf{y}_i}(1-F_i)^{1-\mathbf{y}_i} \\ = {} & \sum_{\mathbf{y}\in\{0,1\}^{|\mathcal{A}|}} min\{C_{fn}\pi_1 \prod_i T_i^{\mathbf{y}_i}(1-T_i)^{1-\mathbf{y}_i}, \\ & C_{fp}\pi_0 \prod_i F_i^{\mathbf{y}_i}(1-F_i)^{1-\mathbf{y}_i}\} \end{aligned} \tag{9}$$

where T_i, F_i are the TP rate and FP rate of acquaintance i respectively. $f(\mathbf{y}) = \frac{C_{fn}\pi_1 \prod_{i=1}^{|\mathcal{A}|}(T_i)^{\mathbf{y}_i}(1-T_i)^{1-\mathbf{y}_i}}{C_{fp}\pi_0 \prod_{i=1}^{|\mathcal{A}|}(F_i)^{\mathbf{y}_i}(1-F_i)^{1-\mathbf{y}_i}}$. $\forall \mathbf{y} \in \{0,1\}^l|\delta(y) = 1$ refers to the combination of decisions which causes the system to raise an alarm and vice versa.

B. Acquaintance Selection Algorithm

To solve such a subset optimization problem, the brute force method is to examine all possible combinations of acquaintances and select the one which has the least overall cost. However, the computation complexity is $O(2^n)$. It is not hard to see that the order of selecting acquaintances does not affect the overall cost. We propose an acquaintance selection algorithm based on a heuristic approach to find an acquaintance set which achieves satisfactory overall cost. In this algorithm, the system always selects the nodes which bring the lowest overall cost.

For the ease of demonstration, We assume the maintenance cost can be written as follow:

$$M(\mathcal{A}) = C_a l = C_a|\mathcal{A}| \tag{10}$$

where C_a is the unit maintenance cost of each acquaintance, which includes the cost of communication, detection assistance, and test messages. Note that any other form of maintenance cost can be easily included into the algorithm.

As shown in Algorithm 1, in the beginning, the acquaintance list is empty. The initial cost is the minimum cost of the

Algorithm 1 Acquaintance Selection $(\mathcal{C}, L_{min}, L_{max})$

```
Require: A set of acquaintance candidates $\mathcal{C}$
Ensure: A set of selected acquaintances $\mathcal{A}$ with minimum length $L_{min}$ and max length $L_{max}$ which brings the minimum overall cost
1: $Quit = false$ //quit the loop if $Quit = true$
2: $\mathcal{A} \Leftarrow \emptyset$
3: $U = min(\pi_0 C_{fp}, \pi_1 C_{fn})$ //initialize the overall cost while there is no acquaintance. $min(\pi_0 C_{fp}, \pi_1 C_{fn})$ is the cost when a node makes a decision without feedback from collaborators
4: while $Quit = false$ do
5:     //select the node that reduces cost most in each iteration
6:     $D_{max} = -MAXNUM$ //initialize the maximum cost reduction to the lowest possible
7:     for all $e \in \mathcal{C}$ do
8:         $\mathcal{A} = \mathcal{A} \cup e$
9:         if $U - R(\mathcal{A}) - M(\mathcal{A}) > D_{max}$ //see Equation (9) and Equation (10) for $R(\mathcal{A})$ and $M(\mathcal{A})$ then
10:            $D_{max} = U - R(\mathcal{A}) - M(\mathcal{A})$
11:            $e_{max} = e$
12:        end if
13:        $\mathcal{A} = \mathcal{A} \setminus e$ //remove $e$ from $\mathcal{A}$
14:    end for
15:    if ($D_{max} > 0$ and $|\mathcal{A}| < L_{max}$) or $|\mathcal{A}| < L_{min}$ then
16:        $\mathcal{A} = \mathcal{A} \cup e_{max}$
17:        $\mathcal{C} = \mathcal{C} \setminus e_{max}$ //remove $e_{max}$ from $\mathcal{C}$
18:        $U = U - D_{max}$
19:    else
20:        $Quit = true$
21:    end if
22: end while
```

decision based only on the prior information (line 3). For each loop, the system selects a node from the acquaintance candidate list which brings the lowest overall cost and stores it into e_{max} (lines 7-14), where $U - R(\mathcal{A}) - M(\mathcal{A})$ is the amount of cost reduced by adding a node into the acquaintance list. When such a node is found, it is then moved to the acquaintance list if the current acquaintance length is less than L_{min} or the cost is reduced by adding the new node and the acquaintance length does not exceed L_{max}. The loop stops when no node can be added into $\mathcal{A}$ any further.

C. Acquaintance Management Algorithm

In the previous section, we devised an algorithm to select acquaintances from a list of candidates. However, collaboration is usually based on mutual consensus. If node A selects B as an acquaintance but B does not select A (non-symmetric selection), then the collaboration is not established.

We propose a distributed approach for a HIDS in the CIDN to select and manage acquaintances and a consensus protocol to allow a HIDS to deal with the non-symmetric selection problem. To improve the stability of the acquaintance list, we propose to use a probation period on each new node for the HIDS to learn about the new node before considering it as an acquaintance. For this purpose, each HIDS maintains a *probation list*, where all new nodes remain during their probation periods. A node also communicates with nodes in its probation list periodically to evaluate their detection accuracy. The purpose of the probation list is thus to explore potential collaborators and keep introducing new qualified nodes to the acquaintance list.

Algorithm 2 Managing Acquaintance & Probation Lists

```
1: Initialization :
2: $\mathcal{A} \Leftarrow \emptyset$ //Acquaintance list.
3: $\mathcal{P} \Leftarrow \emptyset$ //Probation list.
4: $l^p = l^{ini}$ //initial Probation length
5: //Fill $\mathcal{P}$ with randomly selected nodes
6: while $|\mathcal{P}| < l^p$ do
7:     $e \Leftarrow$ select a random node
8:     $\mathcal{P} \Leftarrow \mathcal{P} \cup e$
9: end while
10: set new timer event($t_u$, "SpUpdate")
11: Periodic Maintenance:
12: at timer event ev of type "SpUpdate" do
13: //Merge the first mature node into the acquaintance list.
14: $e \Leftarrow selectOldestNode(\mathcal{P})$
15: $\mathcal{C} \Leftarrow \mathcal{A}$ //$\mathcal{C}$ is the temporary candidate list
16: if $t_e > t_p$ //$t_e$ is the age of node $e$ in the probation list then
17:     $\mathcal{P} \Leftarrow \mathcal{P} \setminus e$
18:     if $T_e > T_{min}$ and $F_e < F_{max}$ //$T_e$ and $F_e$ are the true positive rate and false positive rate of the node $e$ then
19:         $\mathcal{C} \Leftarrow \mathcal{C} \cup e$
20:     end if
21: end if
22: //Consensus protocol
23: $\mathcal{S}$ =Acquaintance Selection$(\mathcal{C}, l^{min}, \max(l^{min}, \frac{q}{q+1} l^{max}))$
24: //Send requests for collaboration and receive responses
25: $S_{accp} \Leftarrow RequestandReceiveCollaboration(S, t_{timeout})$
26: $\mathcal{A} \Leftarrow S_{accp}$ //Only nodes that accept the collaboration invitations are moved into the acquaintance list
27: //Refill $\mathcal{P}$ with randomly selected nodes
28: while $|\mathcal{P}| < max(q|\mathcal{A}|, l^{min})$ do
29:     $e \Leftarrow$ Select a random node not in $\mathcal{A}$
30:     $\mathcal{P} \Leftarrow \mathcal{P} \cup e$
31: end while
32: set new timer event($t_u$, "SpUpdate")
33: end timer event
```

Suppose that node i has two sets $\mathcal{A}_i$ and $\mathcal{P}_i$, which are the acquaintance list and the probation list respectively. The corresponding false positive rate and true positive rate of both sets are $F_i^{\mathcal{A}}, T_i^{\mathcal{A}}$ and $F_i^{\mathcal{P}}, T_i^{\mathcal{P}}$. To keep learning the detection accuracy of the acquaintances, a node sends test messages to nodes in both the acquaintance list and the probation list periodically, and keeps updating their estimated false positive rates and true positive rates. Let l^{max} be the maximum number of HIDSes in both the acquaintance and the probation list. We set this upper-bound because the amount of resources used for collaboration is proportional to the number of acquaintances it manages. l^{max} is determined by the resource capacity of each

HIDS. Let l^{min} be the minimum length of a probation list and q be the parameter that controls the length of the probation list l^p compared to the length of acquaintance list l^a, such that $l^{min} \leq l^p \leq ql^a$. The parameters l^{min} and q are used to tune the trade-off between the adaptability to the situation where nodes join or leave the network frequently ("high churn rate"), and the overhead of resources used on testing new nodes.

The acquaintance management procedure for each node is shown in Algorithm 2. The acquaintance list $\mathcal{A}$ is initially empty and the probation list $\mathcal{P}$ is filled by l^{ini} random nodes to utilize the resources in exploring new nodes. An acquaintance list updating event is triggered every t_u time units. $\mathcal{A}$ is updated by including new trusted nodes from $\mathcal{P}$. A node that stays at least t_p time units in probation is called a *mature node*. Only mature nodes are allowed to join the acquaintance list (lines 15-21). Mature nodes with bad qualification will be abandoned right away. After that the acquaintance selection algorithm is used to find the optimal candidate list. Collaboration requests are sent out for nodes which are selected in the optimal list. If an acceptance is received before expiration time then the collaboration is confirmed, otherwise the node is abandoned (lines 22-26). Then, $\mathcal{P}$ is refilled with new randomly chosen nodes (lines 28-31).

Several properties are desirable for an effective acquaintance management algorithm, including convergence, stability, robustness, and incentive-compatibility for collaboration. When our acquaintance management is in place, we are interested to know with whom the HIDS nodes end up collaborating with and how often they change their collaborators. We list potential malicious attacks against our acquaintance management system and present corresponding defense strategies in Section V. We also expect to see cooperative nodes are rewarded and dishonest nodes penalized.

In Section VI we evaluate our acquaintance management algorithm, to determine whether it achieves the above properties.

V. Common Threats and Defense

Efficient collaborative detection can improve detection ability of the whole group. However, the collaboration management itself may become the target of attacks and be compromised. For example, an HIDS may be compromised and turn to be a malicious insider, and then disseminate false information to other HIDSes in the network. An effective CIDN design should be robust to common insider attacks. In this section, we describe common attacks to CIDN and provide defense mechanisms against them.

1) Sybil attacks: occur when a malicious peer in the system creates a large amount of pseudonyms (fake identities) [10]. This malicious peer uses fake identities to gain larger influence over the false diagnosis in the network. For example, fake nodes can gain the trust of some victims and then send them false diagnoses to cause false negative decisions. Our defense against sybil attacks relies on an authentication mechanism. Authentication makes registering fake identities difficult. In our model, the registration of new user IDs requires puzzle solving (such as CAPTCHA) which requires human intelligence to handle. In this way, creating a large number of fake IDs is not practical for an attacker. In addition, our collaboration model requires HIDSes to allocate resources to answer diagnosis requests from their collaborators before being able to influence the collaborators, which is costly to do with many fake identities. This way, our CIDN protects the collaborative network from sybil attacks.

2) Newcomer attacks: occur when a malicious peer can easily register as a new user [27]. Such a malicious peer creates a new ID for the purpose of erasing its bad history with other peers in the network. Our model handles this type of attack by requiring probation period for all newcomers. It takes a long time (for example, more than 10 days) for a new node to gain trust and before they pass probation period, their feedback is simply not considered by other peers. In this way, it takes a long time for a node to gain influence again by registering a new ID. Therefore, our system resists newcomer attacks.

3) Betrayal attacks: occur when a trusted peer suddenly turns into a malicious one and starts sending false diagnoses. A collaborative detection can be degraded dramatically because of this type of attacks. Our adoption of forgetting factor (Equation 3) enables a faster learning of the malicious behavior (see Section VI-E Figure 14 for evaluation results). Our dynamic acquaintance selection algorithm then quickly removes the malicious node from acquaintance list. It will take quite a while for the malicious node to gain back trust because even it gets the chance to join the probation list again, it needs to stay in the probation list for a certain period of time. This design makes it difficult for this malicious node to continue deceiving or to come back as a collaborator within a short period of time.

4) Collusion attacks: happen when a group of malicious peers cooperate together by providing false diagnosis in order to compromise the network. First, our acquaintance selection algorithm employs random selection of potential collaborators. It will be unlikely for multiple malicious peers to be selected together by one peer as collaborators. In addition, in our system, peers will not be adversely affected by collusion attacks. This is because, in our learning model, each peer relies only on its own experience to detect dishonest peers. Not relying on recommendation or reputation makes the system immune to dishonest third party opinions. This is primarily the reason why we do not consider third party recommendations in the current work. We leave this for future investigation after devising an effective approach for coping with collusion attacks (see Section VIII).

VI. Evaluation

In this section, we describe the conducted experiments to demonstrate the desirable properties of our acquaintance management algorithm. We evaluate the cost efficiency of our Bayesian decision model, cost and time efficiency of the acquaintance selection algorithm, and several desired properties of the acquaintance management algorithm. Each experimental result presented in this section is derived from the average of a large number of replications with an overall negligible confidence interval.

TABLE II
SIMULATION PARAMETERS

Parameter	Value	Description
R	10/day	Test message rate
λ	0.95	Forgetting factor
C_{fp}/C_{fn}	20/100	Unit cost of false positive/negative decisions
C_a	0.01	Maintenance cost of one acquaintance
t_p	10 days	Probation period
t_u	1 day	Acquaintance list update interval
l^{ini}	10	Initial probation length
l^{max}	20	Maximum total number of acquaintances
l^{min}	2	Minimum probation list length
T^{min}	0.5	Minimum acceptable true positive rate
F^{max}	0.2	Maximum acceptable false positive rate
q	0.5	Length ratio of probation to acquaintance list
π_1	0.1	Prior probability of intrusions

A. *Simulation Setting*

We simulate an environment of n HIDS peers collaborating together by adding each other as acquaintances. We adopt two parameters to model the detection accuracy of each HIDS, namely, false positive rate (FP) and false negative rate (FN). Notice that in reality most HIDSes have low FP (< 0.1) and FN is normally in the range of $[0.1, 0.5]$ [26]. This is because false positives can severely damage the reputation of the product, so vendors strive to control their FP rate at a low level. In our experiment, we select parameters which reflect real world properties. To test the detection accuracy of acquaintances, each peer sends test messages where their correct answers are known beforehand. Test messages are sent following a Poisson process with average arrival rate R. R will be determined in the next subsection. We use a simulation day as the time unit in our experiments. The diagnosis results given by a HIDS are simulated following a Bernoulli random process. If a test message represents a benign activity, the HIDS i raises alarm with a probability of FP$_i$. Similarly, if the test message represents intrusions, an alarm will be raised with a probability of 1-FN$_i$. All parameter settings are summarized in Table II.

B. *Determining the Test Message Rate*

The goal of our first experiment is to study the relationship between test message rates and FP, FN learning speed. We simulate two HIDSes A and B. A sends B test messages to ask for diagnosis, and learns the FP and FN of B based on the quality of B's feedback. The learning procedure follows Equations (1), (2), and (3). We fix the FN of B to 0.1, 0.2, and 0.3 respectively. Under each case, we run the learning process under different test message rates, 2/day, 10/day, and 50/day respectively. We observe the change of estimated FN over time, plotted in Figure 3. We see that when R is 2/day, the estimated FN converges after around 30 days in the case of FN=0.2. The converging time is slightly longer and shorter in the cases of FN=0.3 and FN=0.1, respectively. When R is increased to 10/day, the converging time decreases to around 10 days. In the case of R=50/day, the corresponding converging time is the shortest (around 3 days) among the three cases. Increasing the test message rate R to 50/day does not reduce much learning process time. Based on the above observation, we choose R=10/day and the probation period t_p to be 10 days as our system parameters. In this way, the test message rate is kept low and the learned FN and FP values converge after the probation period.

The second experiment is to study the efficiency of learning results after our chosen probation period. We fix R=10/day, t_p=10/day, and randomly choose FN of node B uniformly among [0, 1]. We repeat the experiments 100 times with different FNs. The FNs estimated using our learning process till the end of probation period are plotted in Figure 4. We can see that in all different settings of FNs, the estimated FN rates are close to the actual FN rates after the probation period.

C. *Efficiency of our Feedback Aggregation*

In this experiment, we evaluate the effectiveness of our Bayesian decision based feedback aggregation by comparing it with a threshold based aggregation. We have described our Bayesian decision model in Section III-B. In a simple threshold based feedback aggregation method, if the number of HIDSes reporting intrusions is larger than a predefined threshold, then the system raises an alarm. The threshold-based decision is used in N-version cloud anti-virus systems [26].

We set up eight HIDSes $\{\text{HIDS}_0, \text{HIDS}_1, ..., \text{HIDS}_7\}$ with their FP and FN rates randomly chosen from the range [0.1, 0.5]. HIDS$_0$ sends consultations to all other HIDSes, collects and aggregates feedback to make intrusion decisions. The costs of false positive and false negative decisions are C_{fp}=20 and C_{fn}=100 respectively. We compare the average false detection cost using the Bayesian decision model and the simple threshold-based approach. Figure 5 shows that the cost of threshold decision largely depends on the chosen threshold value. An appropriate threshold can significantly decrease the cost of false decisions. In contrast, the Bayesian decision model does not depend on any threshold setting and prevails over the threshold decision under all threshold settings. This is because the threshold decision treats all participants equally, while the Bayesian decision method recognizes different detection capabilities of HIDSes and takes them into account in the decision process. For example, if a HIDS asserts that there is intrusion, our Bayesian model may raise an alarm if the HIDS has a low FP rate and ignores the warning if the HIDS has a high FP rate. However, the threshold based decision model will either raise an alarm or not based on the total number of HIDSes which raise warnings and compare it with a predefined threshold, irrespective of the individual that issued the warning.

D. *Cost and the Number of Collaborators*

We define *risk cost* to be the expected cost from false decisions such as raising false alarms (FP) and missing the detection of an intrusion (FN). We show that introducing more collaborators can decrease the risk cost. In this experiment, we study the impact of the number of collaborators on the risk cost. We set up four groups with an equal number of HIDSes. Nodes in all groups have the same FP rate of 0.03, but their FN rates vary from 0.1 to 0.4, depending on the group they are in. Inside each group every node collaborates

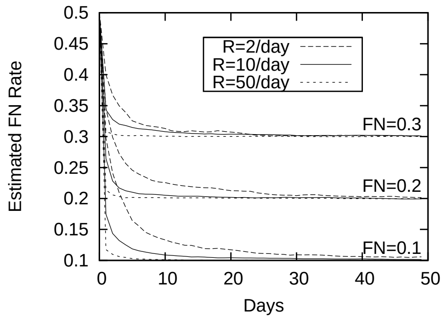

Fig. 3. The Convergence of Learning Speed and the Test Message Rate

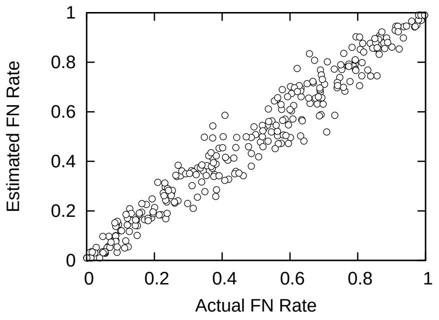

Fig. 4. The distribution of estimated FN rate (R=10/day)

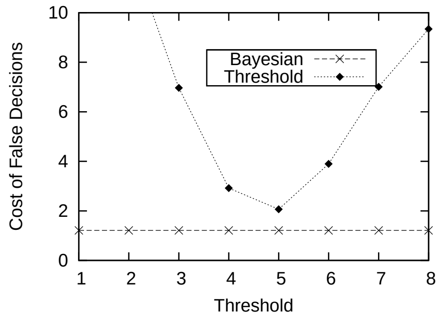

Fig. 5. Comparison of Cost using Threshold Decision and Bayesian Decision

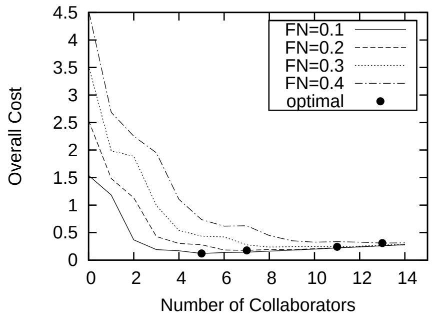

Fig. 6. The Average Cost under Different Collaborator Quality

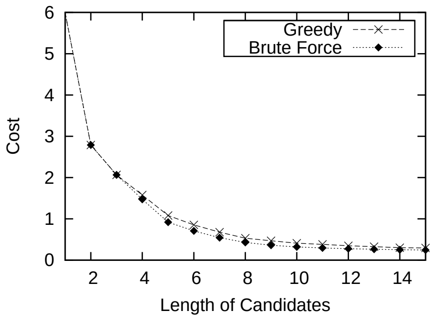

Fig. 7. The Cost using Different Acquaintance Selection Algorithms

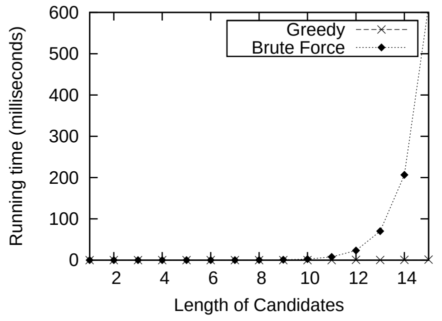

Fig. 8. The Running Time using Different Acquaintance Selection Algorithms

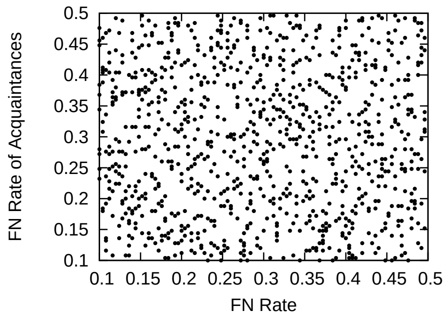

Fig. 9. Acquaintances Distribution on Day 25

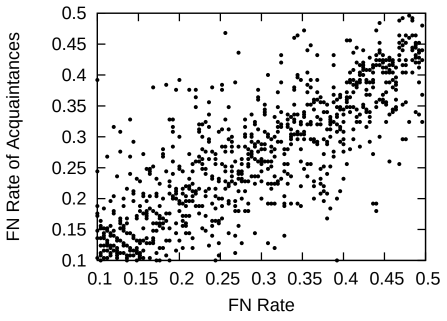

Fig. 10. Acquaintances Distribution on Day 200

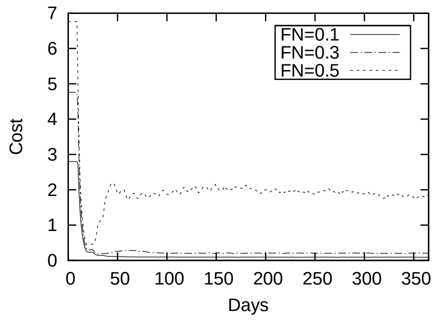

Fig. 11. The Average Cost for Collaboration

with every other node. We are interested in the risk cost as well as the maintenance cost. The *maintenance cost* is the cost associated with the amount of resource that is used to maintain the collaboration with other nodes, such as answering diagnosis requests from other HIDSes. Since our purpose is to capture the concept of maintenance cost but not to study how much it is, we assume the maintenance cost to be linearly proportional to the number of collaborators with a unit rate C_a=0.01 (see Table II).

We increase the size of all groups and observe the average cost of nodes in each group. From Figure 6, we can see that in all groups, the costs drop down fast in the beginning and slow down as the groups' sizes increase. After an optimal point (marked by large solid circles), the costs slowly increase. This is because when the number of collaborators is large enough, the cost saving by adding more collaborators becomes small, and the increment of maintenance cost becomes significant. We find that groups with higher detection accuracy have lower optimal costs. Also they need a smaller number of collaborators to reach the optimal costs. For example, in the case of $FN = 0.4$, 13 collaborators are needed to reach the optimal, while the number of collaborators required is 5 in the case of $FN = 0.1$.

E. *Efficiency of Acquaintance Selection Algorithms*

We learned in the previous section that when the number of collaborators is large enough, adding more collaborators does not decrease the overall cost because of the associated maintenance cost. An acquaintance selection algorithm is proposed in Algorithm 1. In this section, we compare the efficiency of acquaintance selection using the brute force algorithm and our acquaintance selection algorithm. We create 15 HIDSes as candidate acquaintances with FP and FN rates randomly chosen from intervals $[0.01, 0.1]$ and $[0.1, 0.5]$, respectively. Both algorithms are implemented in Java and run on a PC with AMD Athlon dual core processor 2.61GHZ, and with 1.93 GB RAM. We start the candidate set size from 1 and gradually increase the size. We observe the cost efficiency and running time efficiency of both algorithms.

Figure 7 shows that the brute force algorithm performs slightly better with respect to acquaintance list quality since the overall cost using its selected list is slightly lower. However, Figure 8 shows that the running time of the brute force method increases significantly when the candidate set size exceeds 11, and continues to increase exponentially, while our algorithm shows much better running time efficiency. These experiments suggest to use the brute force method only when the size of candidates list is small (≤ 11). When the candidates list is large, our greedy algorithm should be used to select acquaintances.

F. Evaluation of Acquaintance Management Algorithm

In this experiment, we study the effectiveness of our acquaintance management algorithm (Algorithm 2). We set up a simulation environment of 100 nodes. For the convenience of observation, all nodes have fixed FP rate 0.1 and their FN rates are uniformly distributed in the range of $[0.1, 0.5]$. All nodes update their acquaintance list once a day (t_u=1). We observe several properties: convergence, stability, robustness, and incentive-compatibility.

1) Convergence: Our first finding about our acquaintance management algorithm is that HIDSes converge to collaborating with other HIDSes with similar detection accuracy levels. We observed through experiments that HIDSes collaborate with random other nodes in the network in the beginning (Figure 9). After a longer period of time (200 days), all HIDSes collaborate with others with similar detection accuracy, as shown in Figure 10. Our explanation is that the collaboration between pairs with high qualification discrepancy is relatively not stable since our collaboration algorithm is based on mutual consensus and consensus is hard to reach between those pairs.

Figure 11 plots the average overall cost in the first 365 days of collaboration for three nodes with FN values $0.1, 0.3$, and 0.5 respectively. In the first 10 days, the costs for all nodes are high. This is because all collaborators are still in probation period. After day 10, all cost values drop down significantly. This is because collaborators pass probation period and start to contribute to intrusion decisions. The cost for high expertise nodes continues to drop while the cost for low expertise nodes increases partially after around day 20, and stabilizes after day 50. This is because the acquaintance management algorithm selects better collaborators to replace the initial random ones. We can see that the collaboration cost of nodes converges with time and becomes stable after the initial phase.

2) Stability: Collaboration stability is an important property since the collaboration between HIDSes is expected to be long term. Frequently changing collaborators is costly because HIDSes need to spend considerable amount of time to learn about new collaborators. In this experiment, we record the average time span of all acquaintances from the time they pass probation period till they are replaced by other acquaintances. The result is shown in Figure 12, where the average collaboration time spans for three selected nodes are shown with different point shapes. We can see that collaboration among nodes with similar expertise levels is more stable than that between nodes with different expertise levels. For example, nodes with low $FN = 0.1$ form stable collaboration connections with other nodes with low FN (around 180 days in average), while the collaboration with HIDSes with high FN is short (close to 0 day in average).

3) Incentive-compatibility: Collaboration among HIDSes is expected to be a long term relationship. Incentive is important for the long term sustainability of collaborations since it provides motivation for peers to contribute [12], [9]. We compare the average overall cost of all nodes with different FN rates under three different conditions, namely, no collaboration, fixed acquaintances collaboration (acquaintance length=8), and dynamic acquaintance management collaboration. Figure 13 shows the distribution of the converged cost of all nodes. We can observe that the cost of all HIDSes is much higher when no collaboration is performed in the network. On the other hand, collaborating with random fixed acquaintances can significantly reduce the cost of false decisions, however, the cost of high expertise nodes and low expertise nodes are very close. With our dynamic acquaintance management, high expertise nodes achieve much lower cost than nodes with low expertise, which reflects an incentive design of the collaboration system. Therefore, the system provides motivation for nodes to update their knowledge base and behave truthfully in cooperation.

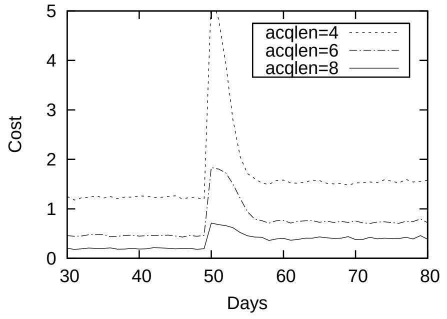

Fig. 15. The Cost of a HIDS under a Betrayal Attack

4) Robustness: Robustness is a desired property of a CIDN since malicious users may try to attack the collaboration mechanism to render it ineffective. Several strategies can be adopted as we discussed in Section V. We focus on the Betrayal attack in this experiment. To study the impact from one malicious node, we set up a collaboration scenario where HIDS_0 is collaborating with a group of other HIDSes with $FP = 0.1$ and $FN = 0.2$. Among the group, one HIDS turns to be dishonest after day 50 and gives false diagnoses. We observe the FP rate and FN rate of this malicious node perceived by HIDS_0, and the impact on the risk cost of HIDS_0 under various collaborator group sizes. Figure 14 shows the perceived FP

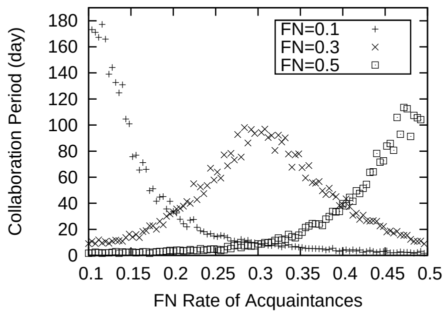

Fig. 12. The Collaboration Time Span

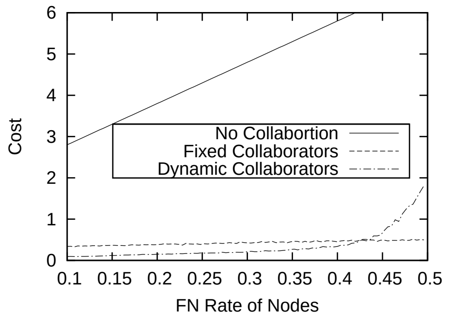

Fig. 13. The Converged Cost Distribution

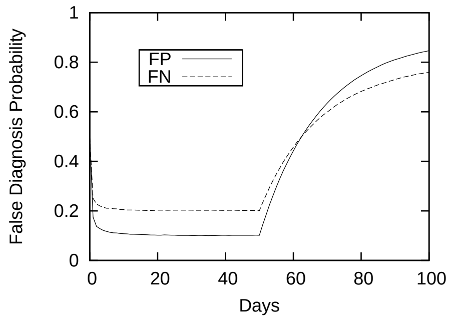

Fig. 14. The FP and FN of Betrayal Node

and FN rate of the malicious node during each simulation day. We can see that the perceived FP and FN increase fast after day 50. The malicious node is then removed from the acquaintance list of $HIDS_0$ when its perceived FP and FN are higher than a predefined threshold. The cost of $HIDS_0$ under betrayal attack is depicted by Figure 15; we notice that the betrayal behavior introduces a spike of cost increment under all group sizes, but the magnitude of increment decreases when the number of collaborators increases. However, the system can efficiently learn the malicious behavior and recover to normal by excluding malicious nodes from the acquaintance list.

VII. Related Work

Existing CIDNs can be divided into information-based CIDNs and consultation-based CIDNs. In information-based CIDNs, intrusion information such as intrusion alerts, intrusion traffic samples, firewall logs, and system logs are shared in the network and aggregated to achieve better network-wide intrusion decisions. Several information-based CIDNs, such as [3], [31], [7], and [33], have been proposed in the past few years. They are especially effective in detecting epidemic worms or attacks, and most of them require homogeneous participant IDSes. In contrast, in consultation-based CIDNs, suspicious file samples or traffic samples are sent to collaborators for diagnosis. Diagnosis results (feedback) from collaborators are then aggregated to help the sender IDS make intrusion decisions. Examples of such CIDNs include [14], [15], [13], [11], and [26]. Consultation-based CIDNs may involve heterogeneous IDSes and are effective in detecting many intrusion types including worms, malware, port scanning, and vulnerability explorations. The CIDN proposed in this paper is a consultation-based CIDN. With respect to the CIDN collaboration topology, many proposals are centralized, such as [3], [8], and [31]. A centralized CIDN requires a continuously available central server and it suffers from the single point of failure problem. A decentralized CIDN such as [33], [23] can alleviate the workload of the central server by means of clustering where the cluster heads partially process and summarize the data they collect, and then forward it to higher level nodes for processing. In a fully distributed CIDN [37], [13], [7], [36], all IDSes are equally responsible for collecting/processing information and therefore is the most scalable and flexible topology. The CIDN proposed in this paper is a fully distributed overlay network.

Various approaches have been proposed to evaluate HIDSes. All use a single trust value to measure whether a HIDS will provide good feedback about intrusions based on past experience with this HIDS. For example, Duma et al. [11] introduce a trust-aware collaboration engine for correlating intrusion alerts. Their trust management scheme uses each peer's past experience to predict others' trustworthiness. Our previous work [14], [15] uses Dirichlet distributions to model peer trust, but it does not investigate the conditional detection accuracy such as false positives and false negatives. In this work, we use both false positive and true positive rates to represent the detection accuracy of a HIDS based on a Bayesian learning approach. The methods for aggregating feedback provided by Duma et al. [11] and our previous work [14], [15] are also simplistic. They both use a weighted average approach to aggregate feedback. Another broadly accepted decision model in CIDNs is the threshold-based, which is used in AVCloud [26]. In this model, when the total number of collaborators raising alarms exceeds a fixed threshold, an alarm will be raised. In this paper, we apply the well established Bayes' theorem for feedback aggregation which achieves better performance. Our previous work [14], [15] focuses on the trust evaluation with a simple threshold-based acquaintance selection. This work focuses on the optimal collaboration decision and optimal acquaintance selection.

Most previous approaches set a fixed length of the acquaintance list, such as in [34]. Others use a trust threshold to filter out less honest acquaintances [35], [14]. The advantage of the threshold based decision is its simplicity and ease of implementation. However, it is only effective in a static environment where collaborators do not change, such as that presented in [26]. In a dynamic environment, nodes join and leave the network and the acquaintance list changes with time. Therefore, finding an optimal threshold is a difficult task. Our Bayesian decision model is efficient and flexible. It can be used in both static and dynamic collaboration environments. Equipped with this Bayesian decision model, our acquaintance selection algorithm can find the smallest set of best acquaintances that can maximize the accuracy of intrusion detection. Based on this acquaintance selection algorithm, our acquaintance management method uses a probation list to explore potential candidates for acquaintances and balances the cost of exploration and the speed of updating the acquaintance list.

VIII. Conclusion and Future Work

We proposed a statistical model to evaluate the tradeoff between the maintenance cost and intrusion cost, and an effective acquaintance management method to minimize the overall cost for each HIDS in a CIDN. Specifically, we adopted a Bayesian learning approach to evaluate the accuracy of each HIDS in terms of its false positive and true positive rates in detecting intrusions. The Bayes' theorem is applied for the aggregation of feedback provided by the collaborating HIDSes. Our acquaintance management explores a list of candidate HIDSes and selects acquaintances using an acquaintance selection algorithm. This algorithm is based on a greedy approach to find the smallest number of best acquaintances and minimize the cost of false intrusion decisions and maintenance. The acquaintances list is updated periodically by introducing new candidates which pass the probation period.

Through a simulated CIDN environment, we evaluated our Bayesian decision model against threshold-based decision models, and acquaintance selection algorithm against a brute force approach. Compared to the threshold-based model, our Bayesian decision model performs better in terms of cost of false decisions. Compared to the brute force approach, our algorithm achieves similar performance but requires much less computation time. Our acquaintance management is also shown to achieve the desirable properties of convergence, stability, robustness, and incentive-compatibility.

As future work, we intend to seek a theoretical method to determine the optimal length of the acquaintance list for a HIDS based on the detection accuracy of a set of candidate acquaintances. We will also investigate other more sophisticated attack models on the collaboration mechanism and integrate corresponding defense techniques. Robustness of the acquaintance management system is particularly critical if extended to support HIDS peer recommendations. In this case, malicious HIDSes may provide untruthful recommendations about other HIDSes [32], [35], [24], or worse collide to collaboratively bring the system down.

References

[1] Bro Intrusion Detection System. http://www.bro-ids.org/ [Last accessed in Oct 1, 2011].

[2] OSSEC. http://www.ossec.net [Last accessed in Oct 1, 2011].

[3] SANS Internet Storm Center (ISC). http://isc.sans.org/ [Last accessed in Oct 1, 2011].

[4] Snort Intrusion Detection System. http://www.snort.org/ [Last accessed in Oct 1, 2011].

[5] TripWire. http://www.tripwire.org/ [Last accessed in Oct 1, 2011].

[6] ZDnet. http://www.zdnet.com/blog/security/confickers-estimated-economic-cost-91-billion/3207 [Last accessed in Oct 1, 2011].

[7] M. Cai, K. Hwang, Y. Kwok, S. Song, and Y. Chen. Collaborative Internet Worm Containment. *IEEE Security & Privacy*, 3(3):25–33, 2005.

[8] F. Cuppens and A. Miege. Alert Correlation in a Cooperative Intrusion Detection Framework. In *IEEE Symposium on Security and Privacy.*, 2002.

[9] A. Dal Forno and U. Merlone. Incentives and Individual Motivation in Supervised Work Groups. *European Journal of Operational Research*, 207(2):878–885, 2010.

[10] J. Douceur. The Sybil Attack. *Peer-To-Peer Systems: First International Workshop, IPTPS*, 2002.

[11] C. Duma, M. Karresand, N. Shahmehri, and G. Caronni. A Trust-Aware, P2P-Based Overlay for Intrusion Detection. In *DEXA Workshops*, 2006.

[12] E. Fehr and H. Gintis. Human Motivation and Social Cooperation: Experimental and Analytical Foundations. *Annu. Rev. Sociol.*, 33:43–64, 2007.

[13] C. Fung, O. Baysal, J. Zhang, I. Aib, and R. Boutaba. Trust Management for Host-based Collaborative Intrusion Detection. In *19th IFIP/IEEE International Workshop on Distributed Systems*, 2008.

[14] C. Fung, J. Zhang, I. Aib, and R. Boutaba. Robust and Scalable Trust Management for Collaborative Intrusion Detection. In *Proceedings of the Eleventh IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2009.

[15] C. Fung, J. Zhang, I. Aib, and R. Boutaba. Dirichlet-based Trust Management for Effective Collaborative Intrusion Detection Networks. *IEEE Transactions on Network and Service Management (TNSM)*, 8(2):79–91, 2011.

[16] C. J. Fung, J. Zhang, and R. Boutaba. Effective Acquaintance Management for Collaborative Intrusion Detection Networks. In *6th International Conference on Network and Service Management (CNSM)*, 2010.

[17] A. Gelman. *Bayesian data analysis*. CRC press, 2004.

[18] T. Holz, C. Gorecki, K. Rieck, and F. Freiling. Detection and Mitigation of Fast-Flux Service Networks. In *Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS)*, 2008.

[19] R. Janakiraman and M. Zhang. Indra: a Peer-to-peer Approach to Network Intrusion Detection and Prevention. *WET ICE 2003. Proceedings of the 12th IEEE International Workshops on Enabling Technologies*, 2003.

[20] A. Jøsang and R. Ismail. The Beta Reputation System. In *Proceedings of the Fifteenth Bled Electronic Commerce Conference*, 2002.

[21] A. Kind, M. P. Stoecklin, and X. Dimitropoulos. Histogram-based Traffic Anomaly Detection. *IEEE Transactions on Network and Service Management (TNSM)*, 6(2):110–121, 2009.

[22] P. Li, M. Salour, and X. Su. A Survey of Internet Worm Detection and Containment. *IEEE Communications Surveys & Tutorials*, 10(1):20–35, 2008.

[23] Z. Li, Y. Chen, and A. Beach. Towards Scalable and Robust Distributed Intrusion Alert Fusion with Good Load Balancing. In *Proceedings of the 2006 SIGCOMM workshop on Large-scale attack defense*. ACM, 2006.

[24] L. Mekouar, Y. Iraqi, and R. Boutaba. A Recommender Scheme for Peer-to-Peer Systems. In *International Symposium on Applications and the Internet (SAINT)*. IEEE, 2008.

[25] J. Mirkovic and P. Reiher. A taxonomy of DDoS attack and DDoS Defense Mechanisms. *SIGCOMM Comput. Commun. Rev.*, 34(2):39–53, 2004.

[26] J. Oberheide, E. Cooke, and F. Jahanian. Cloudav: N-version Antivirus in the Network Cloud. In *Proceedings of the 17th USENIX Security Symposium*, 2008.

[27] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman. Reputation Systems. *Commun. ACM*, 43(12):45–48, 2000.

[28] S. Russell, P. Norvig, J. Canny, J. Malik, and D. Edwards. *Artificial Intelligence: A Modern Approach*, volume 74. Prentice hall Englewood Cliffs, NJ, 1995.

[29] N. Samaan and A. Karmouch. Network Anomaly Diagnosis via Statistical Analysis and Evidential Reasoning. *IEEE Transactions on Network and Service Management (TNSM)*, 5(2):65–77, 2008.

[30] E. Staab, V. Fusenig, and T. Engel. Towards Trust-Based Acquisition of Unverifiable Information. In *Proceedings of the 12th international workshop on Cooperative Information Agents XII (CIA)*, 2008.

[31] J. Ullrich. DShield. http://www.dshield.org/indexd.html [Last accessed in Oct 1, 2011].

[32] P. B. Velloso, R. P. Laufer, D. de O. Cunha, O. C. M. B. Duarte, and G. Pujolle. Trust Management in Mobile Ad Hoc Networks Using a Scalable Maturity-Based Model. *IEEE Transactions on Network and Service Management (TNSM)*, 7(3):172–185, 2010.

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

[34] B. Yu and M. Singh. Detecting Deception in Reputation Management. *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, 2003.

[35] J. Zhang and R. Cohen. Evaluating the Trustworthiness of Advice about Selling Agents in E-Marketplaces: A Personalized Approach. *Electronic Commerce Research and Applications*, 7(3):330–340, 2008.

[36] Z. Zhong, L. Ramaswamy, and K. Li. ALPACAS: A Large-Scale Privacy-Aware Collaborative Anti-Spam System. In *The 27th Conference on Computer Communications (IEEE INFOCOM)*, 2008.

[37] C. Zhou, S. Karunasekera, and C. Leckie. A Peer-to-Peer Collaborative Intrusion Detection System. In *Proceedings of the IEEE International Conference on Networks*, 2005.