Effective Acquaintance Management for Collaborative Intrusion Detection Networks

Carol J Fung
School of Computer Science
University of Waterloo, Canada
j22fung@uwaterloo.ca

Jie Zhang
School of Computer Engineering
Nanyang Technological University, Singapore
zhangj@ntu.edu.sg

Raouf Boutaba
School of Computer Science
University of Waterloo, Canada
rboutaba@uwaterloo.ca

***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. Towards 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. More 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 their opinions about intrusions 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 and incentive of our system.**

I. Introduction

In recent years, cyber attacks from 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 Conflicker worm which infected more than 3 million Microsoft server systems during the year of 2008 to 2009, with the estimated economic lost of 9.1 billion dollars [2]. As a countermeasurement, *Host-based Intrusion Detection Systems* (HIDSes) identify intrusions by comparing observable intrusion data such as log files and computer activities against suspicious patterns. Several examples of HIDSes are OSSEC [1], an antivirus software and Tripwire.

Traditional HIDSes work in isolation and may be easily compromised by unknown or new threats. Collaboration among HIDSes can overcome this weakness by having each peer benefit from the collective knowledge and experience shared by other peers. This enhances the overall accuracy of intrusion assessment as well as the ability of detecting new classes of intrusions. A *Collaborative Intrusion Detection Network* (CIDN) is an overlay software which provides collaboration infrastructure 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 [15], CRIM [4], and N-version AV [12], or distributed, such as Indra [10], and NetShield [3].

However, in a CIDN, malicious insiders may send false information to mislead other HIDSes to make incorrect intrusion decisions, in this way, render the collaboration system not useful. A CIDN *acquaintance management* is the process of identifying, selecting, and maintaining collaborators for each HIDS. Effective acquaintance management is critical to the design of a CIDN. In our paper, we provide a Bayesian learning technique that helps each HIDS to identify dishonest collaborators and remove them from its collaborator list. We propose a Bayesian decision model for HIDSes to aggregate feedback from their collaborators to minimize expected cost of making false decisions. In terms of 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 significant maintenance cost. Previous works on acquaintance management often set a fixed length of collaborators [16], or a threshold to filter out less honest collaborators [17], [9]. These simple approaches lack of flexibility, and the cost efficiency using their selected collaborators largely depends on the context of their networks (i.e. the quality of candidates). Our proposed acquaintance management algorithm can dynamically select collaborators in any context setting to obtain high efficiency on cost. We prove empirically that our acquaintance management algorithm achieves several desired properties, such as efficiency, stability, and incentive.

II. Collaborative Intrusion Detection Network

A 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. Peers may have different expertise levels on intrusion detection. They may also act dishonestly or selfishly. In this paper, we use the terms collaborator and acquaintance interchangeably.

When a HIDS detects suspicious behavior but lacks confidence to make a decision whether it should raise an alarm or not, it may send *consultation requests* to its collaborators for diagnosis. The consultation request contains the description of the suspicious activities or data, such as the data packets from a suspicious source or the log file entries of 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. It is thus important to evaluate peer HIDSes and choose the ones who provide higher detection accuracy.

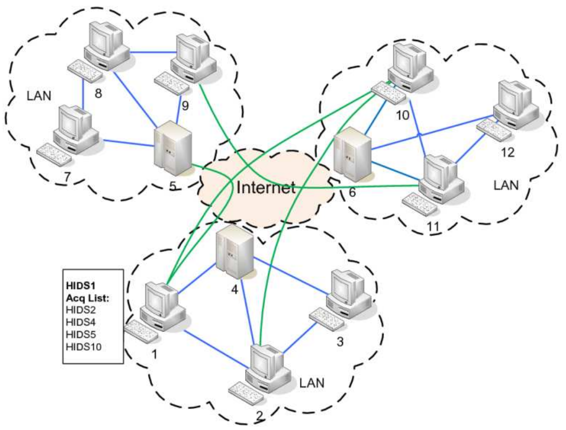

Fig. 1. The Overlay Network of Collaborating HIDSes

In our system, HIDSes also use *test messages* to evaluate the detection accuracy and truthfulness of other HIDSes. Test messages are “bogus” consultation requests, 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 others. This method helps with identifying inexperienced and/or malicious nodes within the network. The idea of “test messages” was previously introduced in [14] and [8]. It is adopted in our CIDN to enable the quick gaining of experience with affordable communication cost.

The evaluation result of acquaintances’ detection accuracy is used in feedback aggregations to achieve better decision accuracy. The acquaintance management then updates acquaintance list periodically to recruit new nodes 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 in detail 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 candidates. In this section, we first introduce a Beta learning model to evaluate the detection accuracy of the candidates. A Bayesian decision model is used to optimally aggregate feedback from acquaintances.

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 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, written as $\mathbb{P}[Y = 1|X = 0]$. TP is the probability that the HIDS gives a correct positive diagnosis under the condition of under-attack, written as $\mathbb{P}[Y = 1|X = 1]$, where random variable $X \in \{0, 1\}$ represents the random event on whether there is an attack or not. Random variable $Y \in \{0, 1\}$ denotes whether the HIDS makes a positive diagnosis or not.

Let $\mathcal{F}_k$ and $\mathcal{T}_k$ be the probability density functions of F_k and T_k whose support is $[0, 1]$. Using the past experience as samples, a Beta distribution function can be used to model the posterior distribution of $\mathcal{F}_k$ and $\mathcal{T}_k$:

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

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

where $\Gamma(\cdot)$ is the gamma function [11], and its parameters α_k^0, α_k^1 and β_k^0, β_k^1 are given by

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

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. $r_{k,j}^0 \in \{0, 1\}$ is the jth diagnosis data 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: $r_{k,0}^1 = 1$ means that the diagnosis from k is positive under attack. $r_{k,0}^1 = 0$ means otherwise. Parameters t_{kj}^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. We use exponential moving average to accumulate past experience so that old experience takes less weight than new expereince.

B. Feedback Aggregation

A node receives a feedback vector $\mathbf{y}$ from its acquaintances. $X \in \{0, 1\}$ denotes the scenario of “no-attack” or “under-attack”. 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 [13] and assuming that the acquaintances provide diagnoses independently and their FP rate and TP rate are known, we have

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

where $\pi_0 = \mathbb{P}[X=0]$ and $\pi_1 = \mathbb{P}[X=1]$ ($\pi_0 + \pi_1 = 1$) are the prior probabilities of the scenarios of "no-attack" and "under-attack". $\mathbf{y}_k$ 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,

$$\begin{aligned}\mathbb{E}[P] &\approx \frac{1}{1+\frac{\pi_0 \prod_{k=1}^{|\mathcal{A}|}\mathbb{E}[F_k]^{\mathbf{y}_k}(1-\mathbb{E}[F_k])^{1-\mathbf{y}_k}}{\pi_1 \prod_{k=1}^{|\mathcal{A}|}\mathbb{E}[T_k]^{\mathbf{y}_k}(1-\mathbb{E}[T_k])^{1-\mathbf{y}_k}}} \\ &= \frac{1}{1+\frac{\pi_0}{\pi_1}\prod_{k=1}^{|\mathcal{A}|}\frac{\alpha_k^1+\beta_k^1}{\alpha_k^0+\beta_k^0}(\frac{\alpha_k^0}{\alpha_k^1})^{\mathbf{y}_k}(\frac{\beta_k^0}{\beta_k^1})^{1-\mathbf{y}_k}}. \quad (4)\end{aligned}$$

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,

$$\begin{aligned}R(\delta) &= \int_0^1 (C_{fp}(1-x)\delta + C_{fn}x(1-\delta))f_P(x)dx \\ &= C_{fn}\mathbb{E}[P] + \delta(C_{fp} - (C_{fp}+C_{fn})\mathbb{E}[P]), \quad (5)\end{aligned}$$

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

$$\mathbb{E}[P] \geq \frac{C_{fp}}{C_{fp}+C_{fn}}. \quad (6)$$

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:

$$R(\delta) = \begin{cases} C_{fp}(1-\mathbb{E}[P]) & \text{if } \mathbb{E}[P] \geq \tau, \\ C_{fn}\mathbb{E}[P] & \text{otherwise.}\end{cases} \quad (7)$$

IV. Acquaintance Management

It is intuitive that when a HIDS consults more acquaintances, it achieves higher detection accuracy and lower risk of being compromised. However, having more acquaintances causes higher maintenance cost since the HIDS needs to allocate resource for each node in its acquaintance list. When a HIDS decides 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 used to find the optimal combination 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 $\mathcal{A}_i$. In real applications, maintenance cost of acquaintances may not be negligible since acquaintances send test messages/consultations periodically to ask for diagnosis. It takes resource (CPU and memory) for the host 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 $M_i(\mathcal{A}_i) = C_i \max(|\mathcal{A}_i| - L_i, 0)$, where $L_i \in \mathcal{N}^+$ is the acquaintance length upper-bound and $C_i \in [0,\infty)$ is the penalization of exceeding the bound. 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 all subscript i from our notations for the convenience of presentation. 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 $\pi_0 + \pi_1 = 1$. The above equation can be further written as:

$$\begin{aligned}R(\mathcal{A}) &= C_{fn}\pi_1 \sum_{\forall y \in \{0,1\}^{|\mathcal{A}|}|\delta(y)=0} P[Y=y|X=1] \quad (8) \\ &+ C_{fp}\pi_0 \sum_{\forall y \in \{0,1\}^{|\mathcal{A}|}|\delta(y)=1} P[Y=y|X=0] \\ &= C_{fn}\pi_1 \sum_{\forall y \in \{0,1\}^{|\mathcal{A}|}|\delta(y)=0} \prod_{i=1}^{|\mathcal{A}|}(T_i)^{y_i}(1-T_i)^{1-y_i} \\ &+ C_{fp}\pi_0 \sum_{\forall y \in \{0,1\}^{|\mathcal{A}|}|\delta(y)=1} \prod_{i=1}^{|\mathcal{A}|}(F_i)^{y_i}(1-F_i)^{1-y_i} \\ &= \sum_{y \in \{0,1\}^{|\mathcal{A}|}} min\{C_{fn}\pi_1 \prod_i T_i^{y_i}(1-T_i)^{1-y_i}, \\ &\quad C_{fp}\pi_0 \prod_i F_i^{y_i}(1-F_i)^{1-y_i}\}\end{aligned}$$

where T_i, F_i are the TP rate and FP rate of acquaintance i respectively. $\forall y \in \{0,1\}^l|\delta(y) = 1$ refers to all the combination of decisions which causes the system to raise an alarm, vice versa. Our goal is to select a list 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.

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 to join which bring the lowest overall cost.

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: $Q = 0$ //quit the loop if $Q = 1$
2: $\mathcal{A} \Leftarrow \emptyset$
3: $U = min(\pi_0 C_{fp}, \pi_1 C_{fn})$ //initialize overall cost $R + M$
4: while $Q = 0$ do
5:   //select the node which reduces cost most each time
6:   $D_{max} = -MAXNUM$ //initialize maximum cost
     reduction to lowest possible
7:   for all $e \in \mathcal{C}$ do
8:     $\mathcal{A} = \mathcal{A} \cup e$
9:     if $U - R(\mathcal{A}) - C_a|\mathcal{A}| > D_{max}$ //see Equation (8)
       then
10:      $D_{max} = U - R(\mathcal{A}) - M(\mathcal{A})$
11:      $e_{max} = e$
12:    end if
13:    $\mathcal{A} = \mathcal{A} \setminus e$
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:    $\mathcal{C} = \mathcal{C} \setminus e_{max}$
18:    $U = U - D_{max}$
19:  else
20:    $Q = 1$
21:  end if
22: end while
```

As shown in Algorithm 1, in the beginning, the acquaintance list is empty. The initial cost is the minimum cost of decision based on only the prior information (line 3). For each loop, the system selects a node from the acquaintance candidate list which brings the lowest overall cost (lines 7-14). If such a node is found, it is then moved to the acquaintance list if the acquaintance length is less than L_{min} or the cost is reduced and the acquaintance length is smaller than L_{max}. The loop stops till no node can be added 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.

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: $\mathcal{C} \Leftarrow \mathcal{A}$
16: if $t_e > t_p$ //$t_e$ is the age of node $e$ in probation list then
17:   $\mathcal{P} \Leftarrow \mathcal{P} \setminus e$
18:   if $T_e > T_{min}$ and $F_e < F_{max}$ 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: $S_{accp} \Leftarrow RequestandReceiveCollaboration(S)$
25: $\mathcal{A} \Leftarrow S_{accp}$
26: //Refill $\mathcal{P}$ with randomly selected nodes
27: while $|\mathcal{P}| < max(q|\mathcal{A}|, l^{min})$ do
28:   $e \Leftarrow$ Select a random node not in $\mathcal{A}$ or $\mathcal{P}$
29:   $\mathcal{P} \Leftarrow \mathcal{P} \cup e$
30: end while
31: set new timer event($t_u$, "SpUpdate")
32: end timer event
```

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 from new nodes 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 learn their detection accuracy. The purpose of the probation list is to explore potential collaborators and keep introducing new qualified nodes to the acquaintance list.

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 the estimated false positive rates and true positive rates of them. 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 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 22-25). Then, $\mathcal{P}$ is refilled with new randomly chosen nodes (lines 27-30).

Several properties are desirable for the effective acquaintance management algorithm, including convergence, stability, and incentive 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. Section V evaluates our acquaintance management algorithm, to demonstrate that our algorithm achieves these properties.

V. Evaluation

In this section, we present a set of experiments to demonstrate the desirable properties of our 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.

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 usually in the range of $[0.1, 0.5]$. This is because false positives can severely damage the reputation of the product, so vendors are working hard to minimize their FP rate. 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 rate R. R will be determined in the next subsection. 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 I.

TABLE I
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
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

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 2. 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.

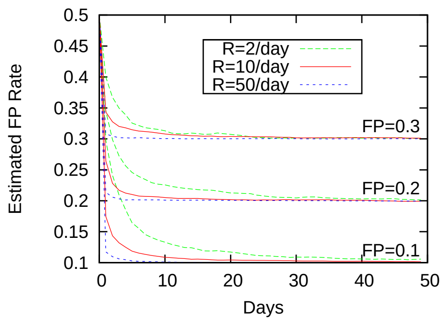

Fig. 2. The Convergence of Learning Speed and the Test Message Rate

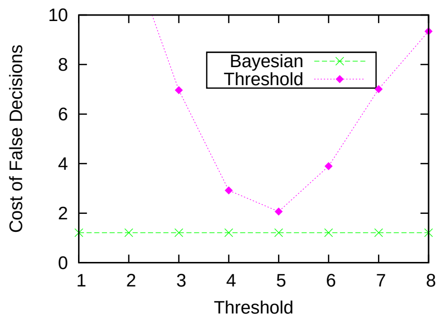

Fig. 3. Comparison of Cost using Threshold Decision and Bayesian Decision

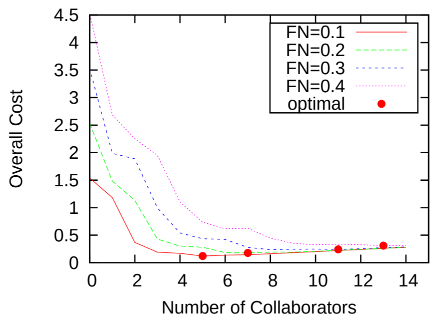

Fig. 4. The Average Cost under Different Collaborator Quality

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 simple 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 [12].

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 and the simple threshold-based decision. Figure 3 shows that the cost of threshold decision largely depends on the threshold value. An appropriate threshold can significantly decrease the cost of false decisions. However, the Bayesian decision prevails 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 based 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 threshold but not on the individual who issued the warning.

D. *Risk Cost and the Number of Collaborators*

Risk cost is 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 equal number of HIDSes. Nodes in all groups have the same FP rate of 0.03, but their FN rates vary from 0.1, 0.2, 0.3, to 0.4, depending on the group they are in. Inside each group every node collaborates with every other node. We are interested in the risk cost as well as maintenance cost. The maintenance cost is the cost associates with the amount of resource that is used to maintain the collaboration with other nodes, such as answering diagnose 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 I).

We increase the size of all groups and observe the average cost of nodes in each group. From Figure 4, we can see that in all groups, the cost drops down fast in the beginning and slowly as group size increases. After an optimal point (marked by large filled circles), the cost slowly increases. We find that groups with higher detection accuracy have lower optimal costs. Also they need a smaller number of collaborators to reach the optimal cost. For example, in the case of FN=0.4, 13 collaborators are needed to reach optimal, while the number of collaborators required is 4 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 IV-B. 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 the 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 run time efficiency of both algorithms.

Figure 5 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 6 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 run time efficiency. Based on our

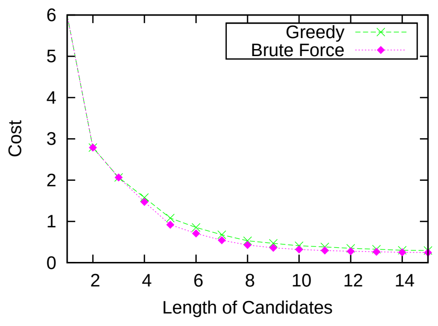

Fig. 5. The Cost using Different Acquaintance Selection Algorithms

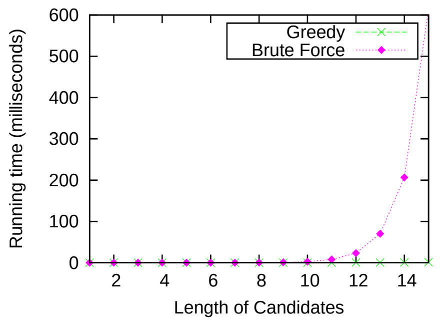

Fig. 6. The Running Time using Different Acquaintance Selection Algorithms

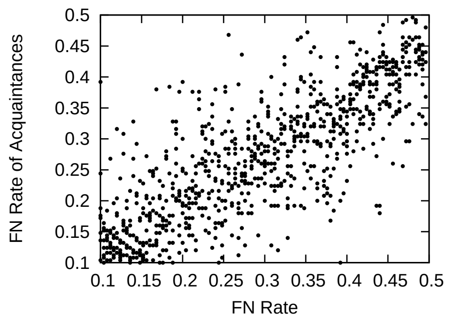

Fig. 7. Acquaintances Distribution on Day 200

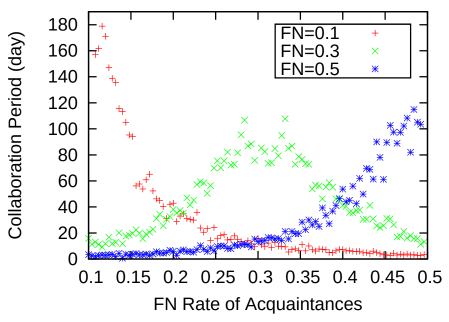

Fig. 8. The Collaboration Time Span

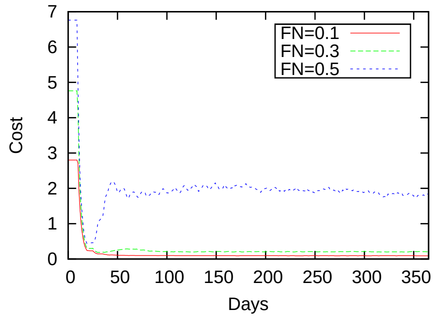

Fig. 9. The average cost for collaboration

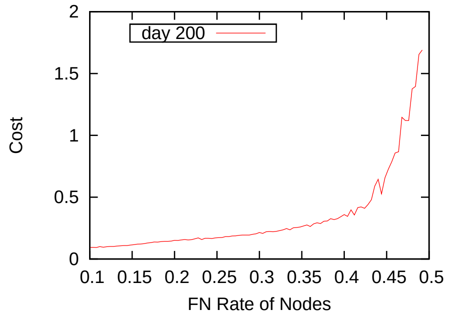

Fig. 10. The Converged Cost Distribution

experiment, we suggest that the brute force method should only be used when the size of candidate list is small (≤ 11). When the candidate 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 are interested to observe several properties, including convergence, stability, and incentive.

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. After a long term (200 days), all HIDSes collaborate with others with similar detection accuracy, as shown in Figure 7. Our reasoning 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.

2) Stability: Collaboration stability is an important peoperty since cooperation between HIDSes is expected to be long term. Frequently changing collaborators is costly because HIDSes need to spend considerable amount of time to learn the honesty of new collaborators. The stability can be observed from Figure 8, where the average collaboration time spans for three selected nodes are shown with different point shapes. We can see that the collaboration among nodes with similar expertise levels is more stable. For example, the node with low FN=0.1 forms very stable collaboration connection with other nodes with low FN (around 180 days), while the collaboration with HIDSes with high FN is very short (close to 0 days).

3) Incentive: The collaboration among HIDSes is a long term collaboration relationship. Incentive is important for the long term sustainability of collaborations since it provides motivation for peers to contribute [6], [7]. Figure 9 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 collaboration can significantly decrease the cost from intrusions for all participants. Figure 10 shows the distribution of the converged cost of all nodes on day 200. We can observe that the HIDSes with higher detection accuracy can achieve less cost in intrusion detection. This provides incentive for nodes to behave truthfully in cooperation.

VI. Related Work

Various approaches have been proposed to evaluate HIDSes. All these approaches 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. [5] 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 [9] uses Dirichlet distributions to model peer trust, but it does not investigate the conditional detection accuracy such as false positives and false negatives. In the current work, we use both the false positive rate and true positive rate to represent the detection accuracy of a HIDS based on a Beta learning approach, to better capture the information about the detection ability of the HIDS. The methods for aggregating feedback provided by Duma et al. [5] and our previous work are also simplistic. Indeed, they both use a weighted average approach to aggregate feedback. Another broadly accepted decision model in CIDN is threshold-based, which is used in AVCloud [12]. In their 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.

Most previous works set a fixed length of the acquaintance list, such as in [16]. Others use a trust threshold to filter out less honest acquaintances [17], [9]. The advantage of the simple threshold based decision is simplicity and ease of implementation. However, it is only effective in a static environment where collaborators do not change, such as the environment presented in [12]. In a dynamic environment, finding the 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 based on a greedy approach can find the smallest number of best acquaintances that can maximize the accuracy of intrusion detection.

VII. Discussion and Future Work

In this paper, we proposed a statistical model to evaluate the tradeoff between the maintenance cost and intrusion cost, and used an effective acquaintance management method to minimize the overall cost for each HIDS in a CIDN. More specifically, we adopted the 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 to the aggregation of feedback provided by the modeled 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 can minimize the cost of intrusion detection. Through a simulated CIDN environment, we evaluated our acquaintance selection algorithm against a brute force approach. Comparatively our algorithm achieves similar performance but requires much less computation time. Our acquaintance management is also demonstrated to be able to achieve the desirable properties of convergence, stability and incentive.

As our future work, we will seek a theoretical method to find the optimal length of the acquaintance list of a HIDS based on the detection accuracy of a set of candidate acquaintances. We will also study attack models intending to compromise the collaboration mechanism and integrate corresponding defense techniques to improve the robustness of our acquaintance management algorithm.

Acknowledgment

This work was supported in part by the Natural Science and Engineering Council of Canada (NSERC) under its Strategic program, Cisco Systems, and in part by the WCU (World Class University) program through the Korea Science and Engineering Foundation funded by the Ministry of Education, Science and Technology (Project No. R31-2008-000-10100-0).

References

[1] OSSEC. http://www.ossec.net [Last accessed in Aug 15, 2010].

[2] ZDnet. http://www.zdnet.com/blog/security/confickers-estimated-economic-cost-91-billion/3207 [Last accessed in Aug 15, 2010].

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

[4] F. Cuppens and A. Miege. Alert correlation in a cooperative intrusion detection framework. In *2002 IEEE Symposium on Security and Privacy, 2002. Proceedings*, pages 202–215, 2002.

[5] C. Duma, M. Karresand, N. Shahmehri, and G. Caronni. A trust-aware, p2p-based overlay for intrusion detection. In *DEXA Workshops*, 2006.

[6] E. Fehr and H. Gintis. Human motivation and social cooperation: Experimental and analytical foundations. 2007.

[7] A. Forno and U. Merlone. Incentives and individual motivation in supervised workgroups. *European Journal of Operational Research*, 2010.

[8] 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.

[9] 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.

[10] 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.

[11] A. Jøsang and R. Ismail. The beta reputation system. In *Proceedings of the Fifteenth Bled Electronic Commerce Conference*, pages 324–337, 2002.

[12] J. Oberheide, E. Cooke, and F. Jahanian. Cloudav: N-version antivirus in the network cloud. In *Proceedings of the 17th USENIX Security Symposium*, 2008.

[13] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach.* Second Edition, Prentice Hall, Englewood Cliffs, New Jersey, 2002.

[14] E. Staab, V. Fusenig, and T. Engel. Towards trust-based acquisition of unverifiable information. In *CIA '08: Proceedings of the 12th international workshop on Cooperative Information Agents XII*, pages 41–54, Berlin, Heidelberg, 2008. Springer-Verlag.

[15] J. Ullrich. DShield. http://www.dshield.org/indexd.html [Last accessed in Aug 15, 2010].

[16] B. Yu and M. Singh. Detecting deception in reputation management. *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 73–80, 2003.

[17] J. Zhang and R. Cohen. Trusting advice from other buyers in e-marketplaces: the problem of unfair ratings. In *ICEC '06*, pages 225–234, New York, NY, 2006.