Design and Management of Collaborative Intrusion Detection Networks

Carol J Fung and Raouf Boutaba
David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada
{j22fung, rboutaba}@uwaterloo.ca

***Abstract*—Traditional intrusion detection systems (IDSs) work in isolation and are not effective to detect unknown threats. An intrusion detection network (IDN) is a collaborative IDS network intended to overcome this weakness by allowing IDS peers to share detection knowledge and experience, and hence improve the overall accuracy of intrusion assessment. However, malicious insiders and free riders may compromise the efficiency of IDNs. In this work, we design a collaborative IDN system and particularly focus on four research problems, namely, trust management, collaborative intrusion decision, resource management, and collaborators selection. We evaluate our design in terms of several desired properties such as efficiency, robustness, scalability and incentive-compatibility.**

***Index Terms*—Intrusion detection, collaborative networks, network security and network management.**

I. Introduction

In recent years *network intrusions* have become a severe threat to the privacy and safety of computer users. Each year billions of malicious *cyber attacks* are reported [9], [21]. Attacks are becoming more sophisticated and stealthy, driven by an "underground economy" [10]. Attacks from the Internet are usually accomplished with the assistance of malicious code (a.k.a. *malware*), such as worms, viruses, Trojan horses, or Spyware. The consequences of large scale attacks can be disastrous. An example is the Conflicker worm which infected more than 3 million Windows servers from year 2008 to 2009, with an estimated economic loss of $9.1 billion [7]. Recent intrusion attacks compromise a large number of hosts to form botnets [20]. Hackers aim not only at harvesting private data and identity information from compromised nodes, but also use the compromised nodes to launch attacks such as distributed denial-of-service (DDoS) attacks.

As a counter measure, *Intrusion Detection Systems* (IDS) are used to identify intrusions by comparing observable behavior against suspicious patterns. IDSs can be categorized into host-based IDSs (HIDSs), which monitor the activities of one computer by tracking system files and logs, and network-based IDSs (NIDSs), which monitor network traffic from/to one or a group of computers. Examples of IDSs include antivirus software [5], Snort [4], Bro [1], tripwire [6], and OSSEC [3].

Traditional IDSs monitor computer activities on a single host or network traffic in a sub-network. They do not have a global view of intrusions and are not effective in detecting fast spreading attacks. Therefore, they are not effective in detecting unknown or new threats. In turn, they can achieve better detection accuracy through collaboration. The emergence of standard information models and communication protocols, such as the Intrusion Detection Message Exchange Format (IDMEF) [2], provides a mean for different IDSs to communicate with each other directly. Accordingly, IDSs are able to exchange information to improve intrusion detection accuracy. An *Intrusion Detection Network* (IDN) is such a collaboration network allowing IDSs to exchange information with each other and to benefit from the collective knowledge and experience shared by others. IDNs enhance the overall accuracy of intrusion assessment as well as the ability to detect new intrusion types.

There are two types of IDNs in the literature: information-based and consultation-based. In an *information-based IDN*, nodes share observations and detection knowledge with other nodes in the network, such as suspicious new attacks. This type of IDNs is effective in detecting fast spreading attacks such as worms. However, it may generate large communication overhead and all exchanged information may not be useful to others. In a *consultation-based IDN*, when an IDS detects suspicious activities but does not have enough confidence to make a decision, it may send *consultation requests* to others in the network. *Feedback* from the collaborators can be used to make a final decision whether it is an intrusion or not. Consultation-based IDNs have much less communication overhead, are more effective in terms of communication efficiency, and are the focus of our work.

An IDN is an effective way to improve intrusion detection accuracy. However, to minimize the impact of malicious insiders, it is important to evaluate the trustworthiness of collaborators, and this can be done through test messages. *Test messages* are "bogus" consultation requests used to measure the trustworthiness of others. They are difficult to distinguish from real consultation requests. The tester node knows the true diagnosis result of the test message and uses the received feedback to derive a trust value for the testee node. This technique can discover inexperienced and/or malicious nodes within the collaboration network.

In our work, we focus on the design of a consultation-based IDN. As shown in Figure 1, a consultation-based IDN is an overlay network of collaborating IDSs. IDSs from different vendors are connected in a peer-to-peer manner. Each IDS selectively maintains a list of collaborators (*acquaintances*) to consult with. IDSs may choose to collaborate with other IDSs with which they have had good experience in the past. An IDS can send consultation requests to ask for diagnosis from its collaborators when it can not make a confident decision. Feedback from collaborators are then integrated to make a final decision.

Building an effective IDN is a challenging task. For exam-

978-3-901882-50-0

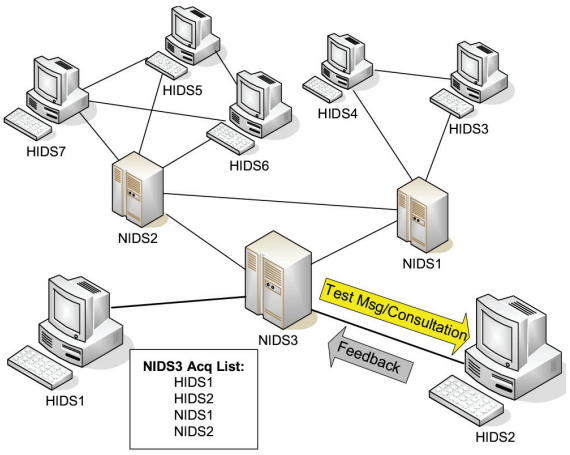

Fig. 1. The Overlay Design of a Collaborative Intrusion Detection Network

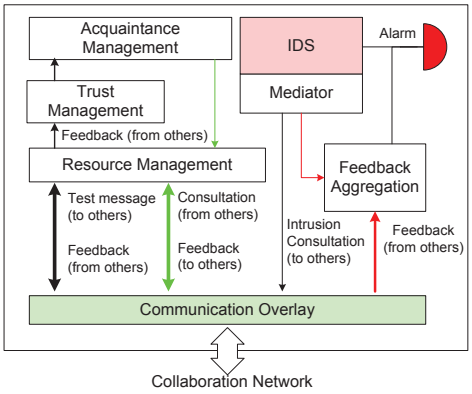

Fig. 2. CIDN Architecture design

ple, adversaries may compromise some IDSs in the network and then leverage the compromised nodes to send false information, or even attack others in the network, which can compromise the efficiency of the IDN. It is, therefore, important for an IDN to detect and isolate malicious insiders. Another challenge is how to make efficient intrusion detection assessment based on the collective diagnosis from other IDSs. Appropriate selection of collaborators and incentive-compatible resource management in support of IDSs interaction with others are also key challenges in IDN design.

Some previous IDS networks such as DOMINO [22], DShield [19], and NetShield [8], focus on the architecture design of the IDS network where nodes share intrusion information to prevent fast spreading attacks. However, they do not address the problems of malicious insiders or free-riders. IDNs such as ABDIAS [17] use a simple majority voting to exclude suspicious nodes, as well as to make collaborative intrusion decisions. However, this simple design can be easily evaded or even taken advantage of by colluding attackers.

To achieve efficiency, robustness, and scalability, we propose an IDN architecture (see Figure 2) which includes several key components, namely, intrusion detection system, mediator, communication overlay, trust management, acquaintance management, resource management, and feedback aggregation. The *mediator* is the component which helps different IDSs to communicate with each other. It translates consultation requests and consultation feedbacks into a common protocol and data format understood by different IDSs. The *communication overlay* is the component which handles all communications with other peers in the collaborative network. It enables IDSs from different vendors to communicate through a common protocol.

In the following sections, we describe the design of these four essential IDN components, namely, trust management, feedback aggregation, resource management, and acquaintance management, which also constitute the core contributions of the thesis.

II. Trust Management for IDN

In a distribute IDN, malicious (or malfunctioning) IDSs can degrade the performance of others by sending false intrusion assessments. To protect an IDN from malicious attacks, it is important to evaluate the trustworthiness of participating IDSs. However, the trust model itself may also be the target of malicious attacks, robustness is a desired feature of the trust management scheme in collaborative intrusion detection networks. In this section, we introduce a fully distributed Bayesian trust model which is scalable, robust, and efficient for intrusion detection networks.

A. Dirichlet-based Trust Model for IDN

Bayesian statistics provide a theoretical foundation for measuring the uncertainty in a decision that is based on a collection of observations. We are interested in knowing the distribution of satisfaction levels of the answers from each peer IDS and, particularly, using this information to estimate the satisfaction level of future consultations. For the case of a binary satisfaction level {satisfied,¬satisfied}, a Beta distribution can be used as appeared in [23]. For multi-valued satisfaction levels, Dirichlet distributions are more appropriate.

A Dirichlet distribution [18] is based on initial beliefs about an unknown event represented by a prior distribution. The initial beliefs combined with collected sample data can be represented by a posterior distribution. The posterior distribution well suits our trust management model since the trust is updated based on the history of interactions.

Let X be the discrete random variable denoting the satisfaction level of the feedback from a peer IDS. X takes values in the set $\mathcal{X} = \{x_1, x_2, ..., x_k\}$ ($x_i \in [0,1]$, $x_{i+1} > x_i$) of the supported levels of satisfaction. Let $\vec{p} = \{p_1, p_2, ..., p_k\}$ ($\sum_{i=1}^{k} p_i = 1$) be the probability distribution vector of X, i.e. $P\{X = x_i\} = p_i$. Also, let $\vec{\gamma} = \{\gamma_1, \gamma_2, ..., \gamma_k\}$ denote the vector of cumulative observations and initial beliefs of X. Then we can model $\vec{p}$ using a posterior Dirichlet distribution as follows:

$$f(\vec{p}|\xi) = Dir(\vec{p}|\vec{\gamma}) = \frac{\Gamma(\sum_{i=1}^{k}\gamma_i)}{\prod_{i=1}^{k}\Gamma(\gamma_i)} \prod_{i=1}^{k} {p_i}^{\gamma_i - 1}, \qquad (1)$$

where ξ denotes the background knowledge, which in here is summarized by $\vec{\gamma}$.

The expected value of the probability of X to be x_i given the history of observations $\vec{\gamma}$ is given by $E(p_i|\vec{\gamma}) = \frac{\gamma_i}{\sum_{i=1}^{k}\gamma_i}$. In

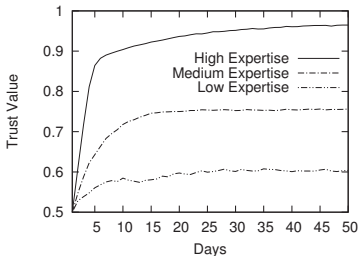

Fig. 3. Convergence of Trust Values for Different Expertise Levels

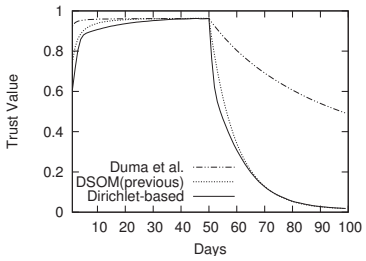

Fig. 4. Trust of Malicious Peers under Betrayal Attack

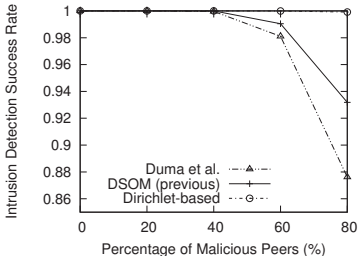

Fig. 5. Intrusion Detection Success Rate under Inconsistency Attack

order to give more weight to recent observations over old ones, we embed a *forgetting factor* λ in the Dirichlet background knowledge vector $\vec{\gamma}$ as follows:

$$\vec{\gamma}^{(n)} = \sum_{i=1}^{n} \lambda^{t_i} \times \vec{S^i} + c_0 \lambda^{t_0} \vec{S^0} \tag{2}$$

where n is the number of observations; $\vec{S^0}$ is the initial beliefs vector. If no additional information is available, all outcomes have an equal probability making $S_j^0 = 1/k$ for all $j \in \{1, .., k\}$. Parameter $c_0 > 0$ is a priori constant, which puts a weight on the initial beliefs. Vector $\vec{S^i}$ denotes the satisfaction level of the i^{th} evidence, which is a tuple containing $k-1$ elements set to zero and only one element set to 1, corresponding to the selected satisfaction level for that evidence. Parameter $\lambda \in [0, 1]$ is the forgetting factor. A small λ makes old observations quickly forgettable. Parameter t_i denotes the time elapsed (age) since the i^{th} evidence $\vec{S^i}$ was observed. Let $\Delta t_i = t_i - t_{i+1}$.

After a peer receives the feedback for a consultation request, it assigns a satisfaction value to the feedback. This satisfaction value is assigned with one of the satisfaction levels in the set $\mathcal{X} = \{x_1, x_2, ..., x_k\}$ that has the closest value. Each satisfaction level x_i also has a weight w_i.

Let p_i^{uv} denote the probability that peer v provides answers to the requests sent by peer u with satisfaction level x_i. Let $\vec{p}^{uv} = (p_i^{uv})_{i=1...k} \mid \sum_{i=1}^{k} p_i^{uv} = 1$. We model $\vec{p}^{uv}$ using Equation 1. Let Y^{uv} be the random variable denoting the weighted average of the probability of each satisfaction level in $\vec{p}^{uv}$.

$$Y^{uv} = \sum_{i=1}^{k} p_i^{uv} w_i \tag{3}$$

The *trustworthiness* of peer v as noticed by peer u is then calculated as:

$$T^{uv} = E[Y^{uv}] = \sum_{i=1}^{k} w_i E[p_i^{uv}] = \frac{1}{\gamma_0^{uv}} \sum_{i=1}^{k} w_i \gamma_i^{uv} \tag{4}$$

where γ_i^{uv} is the cumulated evidence that v has replied to u with satisfaction level x_i. The variance of Y^{uv} is equal to (superscript uv is omitted for clarity).

We evaluated our proposed trust model using a simulated IDS network. Figure 3 shows the average trust values of the 30 IDSs with different expertise levels in the network. The trust values converge after 30 days of simulation and the actual expertise levels of the peers are able to be effectively identified by our trust model.

We also simulated the situation where a malicious peer first gains a high trust value and then suddenly starts to act dishonestly. Figure 4 shows the trust value of the betraying peer before and after the launching of the betrayal attack when respectively using Duma et al., our DSOM model [11] and our Dirichlet models [12] [14]. For the Duma et al. model, the trust value of the malicious peer slowly drops after the betrayal attack. The trust value of the betraying peer drops much faster with the DSOM model, while the fastest rate is observed when using our Dirichlet-based model.

Figure 5 shows the success rate of peer u in detecting intrusions. We notice that both the DSOM model and the Duma et al. model cannot effectively detect intrusions when the majority of peers are malicious. Our Dirichlet-based model shows superior efficiency in intrusion detection even in the situation of a dishonest majority.

B. Summary of Contribution

We proposed a fully distributed Bayesian trust management model that is robust, scalable, and suitable for distributed IDS networks. This trust model provides not only the trust estimation, but also the confidence in the estimation. The forgetting factor parameter is used to balance the learning speed and the stability of the trust value. The full result is published in [12] and [14]. The contributions of this work are two folds: 1) the application of trust modeling in the intrusion detection field, with a model that is robust to insider attacks and scalable to large network sizes. 2) the introduction of Dirichlet density functions in the trust modeling field, which allows the tracking of confidence levels in the estimation of trustworthiness of IDSs.

III. Feedback Aggregation

In a distributed IDN, IDSs can make intrusion decisions based on collected feedback from collaborators. The feedback quality on test messages in the past can be used to evaluate the importance of the current feedback. We derive a decision on whether to raise an alarm or not based on the feedback from a set of collaborators. A false positive decision may cost human resources to investigate it, while a false negative may cause damage to the system. If the decision system is over sensitive then it may bring high false positive alarms, while a conservative decision system may result in high missing rate on intrusions. In this section, we introduce a Baysian decision model which leverages the trade off between false positives and false negatives to find a decision which yields a minimal cost in term of false decisions.

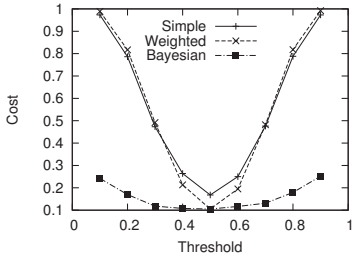

Fig. 6. Average Cost vs. Threshold τ_p

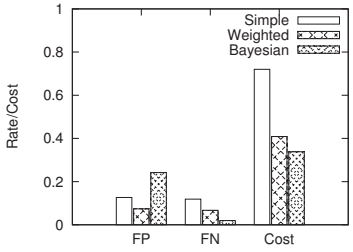

Fig. 7. Comparison of three aggregation models

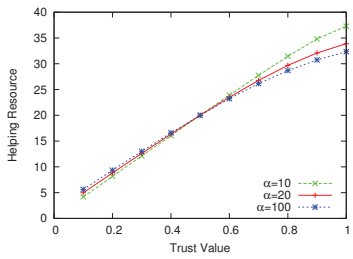

Fig. 8. Resource received varies with trust value

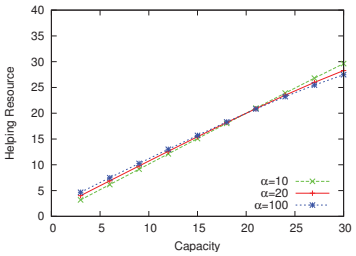

Fig. 9. Resource received varies with resource contribution

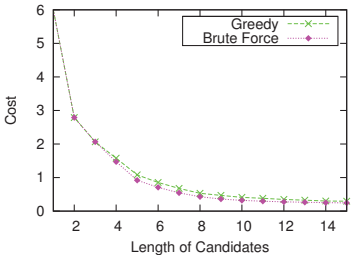

Fig. 10. The Cost using Different Acquaintance Selection Algorithms

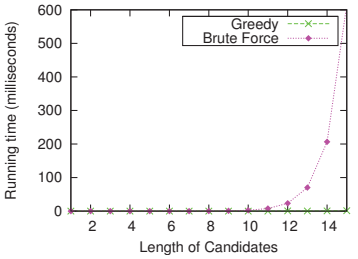

Fig. 11. The Run-time using Different Acquaintance Selection Algorithms

A. Bayesian Decision Model

We formulate the feedback aggregation problem as a Bayesian optimization problem. Consider a set of nodes $\mathcal{N}$ connected to a network, which can be represented by a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$. The set $\mathcal{E}$ contains the undirected links between nodes, indicating the acquaintances of IDSs in the network.

Let $\mathbf{Y}_i := [Y_j]_{j \in \mathcal{A}_i}$ be an observation vector of an IDS i that contains the feedback from its peers in the acquaintance list $\mathcal{A}_i$. For the convenience of presentation, we drop the subscript i in the notations appearing later in this section. Suppose node i receives a list of diagnosis results $\mathbf{y} = \{\mathbf{y}_1, ..., \mathbf{y}_{|\mathcal{A}|}\}$ from its acquaintances, where $\mathbf{y}_j \in \{0, 1\}$, $j = 1, 2, \cdots, |\mathcal{A}|$. $\mathbf{y}_j = 1$ means that the j-th acquaintance suggests an intrusion related to the alert, whereas $\mathbf{y}_j = 0$ indicates no intrusion related to the alert. Our goal is to decide whether the system should raise an alarm to the administrator based on the current feedback.

A node receives a feedback vector $\mathbf{y}$ from its acquaintances. Let random variable $X \in \{0, 1\}$ denote the scenarios of "no-attack" or "under-attack". The probability of a host IDS being "under-attack" given the diagnosis results from all acquaintance IDSs can be written as $\mathbb{P}[X = 1|\mathbf{Y} = \mathbf{y}]$. Using Bayes' Theorem, we have

$$\mathbb{P}[X = 1|\mathbf{Y} = \mathbf{y}] = \frac{\mathbb{P}[\mathbf{Y} = \mathbf{y}|X = 1]\mathbb{P}[X = 1]}{\mathbb{P}[\mathbf{Y} = \mathbf{y}|X = 1]\mathbb{P}[X = 1] + \mathbb{P}[\mathbf{Y} = \mathbf{y}|X = 0]\mathbb{P}[X = 0]}.$$

Assume that the acquaintances provide diagnoses independently and their false positive (FP) and true positive (TP) rates are known; the above equation can be further written as

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

where $\pi_0 = \mathbb{P}[X = 0]$, $\pi_1 = \mathbb{P}[X = 1]$, and $\pi_0 + \pi_1 = 1$, are the prior probabilities of the scenarios of "no-attack" and "under-attack". T_k and F_k are the true positive rates and false positive rates of acquaintance k respectively. $\mathbf{y}_k$ is the k-th element of vector $\mathbf{y}$.

We use a random variable P to denote the conditional probability $\mathbb{P}[X = 1|\mathbf{Y} = \mathbf{y}]$. Then P takes a continuous value over domain $[0, 1]$. We denote by $f_P(p)$ the probability density function of P.

Let C_{fp} and C_{fn} denote the marginal cost of a FP decision and a FN decision. 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,

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

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

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. This decision rule can be written as follows:

$$\delta = \begin{cases} 1 \text{ (Alarm)} & \text{if } \mathbb{E}[P] \geq \tau, \\ 0 \text{ (No alarm)} & \text{otherwise.} \end{cases} \quad (6)$$

We evaluate our Bayesian decision making approach using a simulated IDN. Figure 6 shows the cost comparison between our Bayesian decision model and the other two commonly used models, i.e., simple threshold model and weighted average model. The average cost yielded by Bayesian aggregation remains the lowest among the three under all threshold settings. The costs of the weighted average aggregation and the simple average aggregation are close to each other. Figure 7 shows the comparison of three different aggregation models in terms of their FP, FN, and cost. We notice that the weighted

average model has significant advantage in the FP, FN rates and cost compared to the simple average model. The Bayesian aggregation model has a higher FP and a lower FN compared to the other two models. However, its cost is the lowest among the three. This is because the Bayesian model leverages FP and FN to minimize the overall cost of false decisions.

B. Summary of Contribution

We proposed a Bayesian decision feedback aggregation, which helps in deciding whether to raise an alarm or not based on the past experience and the current diagnosis results from collaborators. Both false positive decision cost and false negative decision cost are taken into account. Compared with other approaches such as the simple average and the weighted average aggregation, our approach reduces the overall cost of false decisions. The detailed description of this contribution was published in [13].

IV. Resource Management in Collaboration

In a distributed IDN, an IDS may receive requests from different peers for consultation. Responding to those requests requires a certain amount of computing resources, such as CPU, memory, and network bandwidth. An IDS may have a limited resource budget to assist other IDSs in the network and cannot satisfy all the requests. An IDS may also free-ride the system or send false intrusion assessments. Therefore, an effective resource allocation scheme is needed for an IDS to decide its response level to requests from neighboring IDSs.

A. Incentive-Compatible Resource Allocation

We consider an IDN with N peers or nodes. We denote the set of nodes by $\mathcal{N} = \{1, 2, \cdots, N\}$. The set of neighbor nodes of peer u is denoted by $\mathcal{N}_u^d$. We can represent the topology of an IDN by a graph $\mathcal{G} := (\mathcal{N}, \mathcal{E})$, where $\mathcal{E}$ is the set of (u, v) pairs in the network. We use r_{vu} to denote the units of resource that node u should allocate in order to serve v with full satisfaction. The minimum acceptable resource from u to v is m_{vu}. Note that r_{vu}, m_{vu} are chosen by node v and informed to node u during negotiation and this negotiation happens in the beginning of the process. Let $p_{uv} \in \mathbb{R}_+$ be the resource that u allocates to v, for every $u, v \in \mathcal{N}$. The parameter p_{uv} is a decision variable of peer u and is private information determined by u and can be measured by v. To satisfy neighbor v, node u should allocate resource to v over the interval $[m_{vu}, r_{vu}]$.

In this system model, we assume that for each node, the trust values of neighbors are given. Let $T_v^u \in [0, 1]$ be the trust value of peer v assessed by peer u, representing how much peer u trusts peer v. The allocated resource p_{uv} from peer u to v is closely related to the trust value T_v^u.

Each peer maximizes its effort to help its neighbor nodes under its capacity constraint C_u, which is dependent on its own resource capacity. Then, resource allocation should satisfy the following capacity constraint:

$$\sum_{v \in \mathcal{N}_u^d} p_{uv} \leq C_u, \text{ for all } u \in \mathcal{N}. \tag{7}$$

We introduce a utility function for each peer to model the satisfaction level of neighbors. The utility function S_{uv} is given by

$$S_{uv} = \frac{\ln\left(\alpha \frac{p_{uv} - m_{vu}}{r_{vu} - m_{vu}} + 1\right)}{\ln(\alpha + 1)}, \tag{8}$$

where $\alpha \in (0, \infty)$ is a system parameter which controls the satisfaction curve and the term $\ln(\alpha + 1)$ in the denominator is the normalization factor. The function S_{uv} is a concave function on its domain under the condition $\alpha > 1$.

Let $U_u : \mathbb{R}_+^{L(u,d)} \to \mathbb{R}_+$ be the peer u's aggregated altruistic utility, where $L(u, d) = \text{card}(\mathcal{N}_u^d)$, the cardinality of the set $\mathcal{N}_u^d$. Let the payoff function, U_u, for u be given by:

$$U_u = \sum_{v \in \mathcal{N}_u^d} w_{uv} S_{uv}, \quad w_{uv} = T_v^u p_{vu}, \tag{9}$$

where w_{uv} is the weight on peer v's satisfaction level S_{uv}, which is the product of peer v's trust value and amount of helping resource allocated to u. A higher weight is applied on peer v's satisfaction level S_{uv} if peer v is better trusted and more generous in providing help to u. In this system, each peer $u \in \mathcal{N}$ in the IDN intends to maximize U_u within its resource capacity. A general optimization problem (OP) can then be formulated as follows:

$$\begin{aligned} \max_{\{p_{uv}, v \in \mathcal{N}_u^d\}} \quad & \textstyle\sum_{v \in \mathcal{N}_u^d} w_{uv} S_{uv} \\ \text{s.t.} \quad & \textstyle\sum_{v \in \mathcal{N}_u^d} p_{uv} \leq C_u \\ & m_{vu} \leq p_{uv} \leq r_{vu}, \forall v \in \mathcal{N}_u^d, \end{aligned} \tag{10}$$

where S_{uv} and w_{uv} are given by (8) and (9), respectively. Notice that this utility function incorporates incentive-compatibility, since nodes with higher trust have higher weight on their satisfaction level compared to others with lower trust.

Every peer in the network is faced with an optimization problem (OP) to solve. We have N independent optimization problems in the form of (OP) for each node. Hence, we can see this as a multi-player game(GP) with payoff function U_u for each peer u. In [25], [24], we prove this N-person game (GP) admits a Nash Equilibrium (NE). We also develop an iterative algorithm to calculate the NE centrally and we compare it with the NE obtained by simulating the network.

Figure 8 shows the incentive-compatibility of the system by varying the trust value of one participant. We see that the resource received by a node increases with the trustworthiness of the node, under different parameter α settings. We then fix the trust values of all nodes to 1.0 and varying the resource capacity of one peer from 3 to 30, we observe in Figure 9 that the amount of resource the peer receives is almost linearly proportional to the resource it contributes to the others. The above experimental results further confirm that our resource allocation mechanism is incentive-compatible.

B. Summary of Contribution

We proposed an incentive-based resource allocation mechanism, where the amount of resources that each IDS allocates to assist its neighbors is proportional to the trustworthiness and the amount of resources allocated by the neighbors to help this

IDS. The contributions of this work are: 1) A mechanism for optimal resource allocation for each peer to maximize its social welfare with a convex utility function; 2) An N-person non-cooperative game model and an iterative primal/dual algorithm to reach the Nash equilibrium; and 3) Incentive compatibility and robustness that is derived from the resource allocation scheme to tackle the "free-riders", dishonest insiders, and DoS attacks. The detailed description of these contributions have been published in [25], [24].

V. Acquaintance List Management

It is intuitive that when an IDS consults more acquaintances, it achieves higher accuracy and confidence in intrusion detection. However, more acquaintances results in higher maintenance cost since the IDS needs to allocate resource for each acquaintance. When an IDS decides how many acquaintances to recruit, both the intrusion risk cost and the maintenance cost (CPU, memory, and Bandwidth) should be taken into account. When recruiting a node as an acquaintance does not decrease the total cost, the node shall not be added into the acquaintance list. However, how to select acquaintances and how many acquaintances to recruit to achieve optimality are crucial questions when building an efficient IDN. In this work, we first define the acquaintance selection problem, then devise a solution for finding the near-optimal combination of acquaintances with respect to the overall cost.

A. Acquaintance Management Algorithm

Let $\mathcal{A}_i$ denote the set of acquaintances of IDS i. Let $M_i(\mathcal{A}_i)$ be the cost for IDS i to maintain the acquaintance set $\mathcal{A}_i$. 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, bandwidth, and memory) for the IDS to receive, analyze the requests, and reply with corresponding answers. The selection of $M_i(.)$ can be user defined on each host. We use $R_i(\mathcal{A}_i)$ to denote the risk cost of missing intrusions and/or false alarms for IDS 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:

$$\begin{aligned} R(\mathcal{A}) = \sum_{y \in \{0,1\}^{|\mathcal{A}|}} min\{ & C_{fn} \pi_1 \prod_i T_i^{y_i} (1 - T_i)^{1 - y_i}, \\ & C_{fp} \pi_0 \prod_i F_i^{y_i} (1 - F_i)^{1 - y_i} \} \end{aligned} \quad (11)$$

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 combinations of decisions which cause the system to raise an alarm and 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 formulate 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.

To solve this 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. Since in most circumstances there is no particular need to select the optimal list of acquaintances and a near-optimal solution is sufficient. We can use a heuristic approach to find an acquaintance set which achieves satisfactory overall cost with much less computation complexity.

In our proposed algorithm [15], a greedy approach is used where an IDS always select other IDSs to join which bring the lowest overall cost. We also propose a distributed algorithm for the IDS to select and manage acquaintances and a consensus protocol to deal with the non-symmetric selection.

We evaluated our acquaintance selection algorithm using a simulated IDN. Figure 10 and 11 are the comparison results between brute force acquaintance selection and greedy acquaintance selection. We can see 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, the running time of the brute force method increases significantly when the candidate set size exceeds 11, and continues to increase exponentially, while the greedy algorithm shows much better run time efficiency.

B. Summary of Contribution

We proposed an acquaintance management algorithm which can dynamically selects collaborators in any context setting to obtain high efficiency at low cost. We also proposed an acquaintance management algorithm to recruit and maintain new candidates for collaboration. We showed empirically that our acquaintance management algorithm achieves several desired properties, such as efficiency, stability, and incentive-compatibility. The detailed description of these contributions have been published in [15], [16].

VI. Conclusion

Building an efficient, robust, and scalable IDN faces many challenges. We first proposed an architecture design of a distributed IDN, which is based on a peer-to-peer communication overlay to allow efficient and scalable information exchange. We then focused on four important research problems in this context, namely, trust management, collaborative decision making, resource management, and acquaintance management, and provided solutions to each of them. We also studied several desired properties of IDNs, such as robustness, scalability, efficiency, incentive-compatibility, and fairness. We evaluated the proposed solutions with respect to those desired properties and compared them with existing ones in the literature.

VII. Final Remarks

The thesis can be downloaded from http://cs.uwaterloo.ca/~j22fung/thesis.pdf. The work presented in this thesis has received the IM 2009 Best Paper Award and the CNSM 2010 Best Student Paper Award.

References

[1] Bro. http://www.bro-ids.org/[Last accessed in Nov 3, 2012].

[2] Intrusion detection message exchange format (idmef). http://www.ietf.org/rfc/rfc4765.txt [Last accessed in Nov 3, 2012].

[3] OSSEC. http://www.ossec.net/[Last accessed in Nov 3, 2012].

[4] Snort. http://www.snort.org/[Last accessed in Nov 3, 2012].

[5] Symantec. http://www.symantec.com/ [Last accessed in Nov 3, 2012].

[6] TripWire. http://www.tripwire.com/[Last accessed in Nov 3, 2012].

[7] ZDnet. http://www.zdnet.com/blog/security/conficker-estimated-economic-cost-91-billion/3207 [Last accessed in Nov 3, 2012].

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

[9] M. Fossi, G. Egan, K. Haley, E. Johnson, T. Mack, T. Adams, J. Blackbird, M. Low, D. Mazurek, D. McKinney, et al. Symantec internet security threat report trends for 2010. *Volume XVI*, 2011.

[10] M. Fossi, E. Johnson, D. Turner, T. Mack, J. Blackbird, D. McKinney, M. Low, T. Adams, M. Laucht, and J. Gough. Symantec report on the underground economy: July 2007 to june 2008. Technical report, Technical Report, Symantec Corporation, 2008.

[11] 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.

[12] 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.

[13] C. Fung, Q. Zhu, R. Boutaba, and T. Barsar. Bayesian Decision Aggregation in Collaborative Intrusion Detection Networks. In *12th IEEE/IFIP Network Operations and Management Symposium (NOMS)*, 2010.

[14] C. J. 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*, 8(2):79 –91, june 2011.

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

[16] C. J. Fung, J. Zhang, and R. Boutaba. Effective acquaintance management based on bayesian learning for distributed intrusion detection networks. *IEEE Transactions on Network and Service Management*, 9(3):320–332, September 2012.

[17] A. Ghosh and S. Sen. Agent-based distributed intrusion alert system. In *Proceedings of the 6th International Workshop on Distributed Computing (IWDC04)*. Springer, 2004.

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

[19] J. Ullrich. DShield. http://www.dshield.org/indexd.html.

[20] R. Vogt, J. Aycock, and M. Jacobson. Army of botnets. In *ISOC Symp. on Network and Distributed Systems Security*, 2007.

[21] P. Wood, M. Nisbet, G. Egan, N. Johnston, K. Haley, B. Krishnappa, T.-K. Tran, I. Asrar, O. Cox, S. Hittel, et al. Symantec internet security threat report trends for 2011. *Volume XVII*, 2012.

[22] 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.

[23] J. Zhang and R. Cohen. Trusting advice from other buyers in e-marketplaces: the problem of unfair ratings. In *Proceedings of the 8th international conference on Electronic commerce: The new e-commerce: innovations for conquering current barriers, obstacles and limitations to conducting successful business on the internet*, 2006.

[24] Q. Zhu, C. Fung, R. Boutaba, and T. Başar. GUIDEX: A game-theoretic incentive-based mechanism for intrusion detection networks. *IEEE Journal on Selected Areas in Communications (JSAC) Special Issue on Economics of Communication Networks & Systems, to appear*, 2012.

[25] Q. Zhu, C. Fung, R. Boutaba, and T. Başar. A game-theoretical approach to incentive design in collaborative intrusion detection networks. In *Proceedings of the International Symposium on Game Theory for Networks (GameNets)*, May, 2009.