Robust and Scalable Trust Management for Collaborative Intrusion Detection

Carol J Fung Jie Zhang Issam Aib Raouf Boutaba
David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada
{j22fung, j44zhang, iaib, rboutaba}@uwaterloo.ca
Technical Report CS-2008-21

***Abstract*—The accuracy of detecting intrusions within an Intrusion Detection Network (IDN) depends on the efficiency of collaboration between the peer Intrusion Detection Systems (IDSes) as well as the security itself of the IDN against insider threats. In this paper, we study host-based IDNs and introduce a Dirichlet-based model to measure the level of trustworthiness among peer IDSes according to their mutual experience. The model has strong scalability properties and is robust against common insider threats, such as a compromised or malfunctioning peer. We evaluate our system based on a simulated collaborative host-based IDS network. The experimental results demonstrate the improved robustness, efficiency, and scalability of our system in detecting intrusions in comparison with existing models.**

I. Introduction

Intrusion Detection Systems (IDSes) identify intrusions by comparing observable behavior against suspicious patterns. They can be network-based (NIDS) or host-based (HIDS). Traditional IDSes work in isolation and may be easily compromised by unknown or new threats. An Intrusion Detection Network (IDN) is a collaborative IDS network intended to overcome this weakness by having each peer IDS 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 intrusion types.

The centralized collaboration of IDSes relies on a central server to gather and analyze alerts. This technique suffers from the classical performance bottleneck and a single point of failure problems. The distributed collaboration of IDSes can avoid these problems. However, in such collaborative environments, a malicious (or malfunctioning) IDS can degrade the performance of others by sending out false intrusion assessments. To protect an IDN from malicious attacks, it is important to evaluate the trustworthiness of participating IDSes, especially when they are host-based.

In this work, we develop a robust Bayesian trust management model that is scalable and suitable for distributed HIDS collaboration. More specifically, we adopt the Dirichlet family of probability density functions in our trust management for estimating the likely future behavior of a HIDS based on its past history. This theoretical model allows us to track the uncertainty in estimating the trustworthiness of the HIDS, which improves the detection accuracy. Our model also offers excellent scalability properties.

We evaluate our system based on a simulated collaborative HIDS network. The HIDSes are distributed and may have different expertise levels in detecting intrusions. A HIDS may also turn malicious due to runtime bugs, having been compromised, having been updated with a faulty new configuration, or having been deliberately made malicious by its owner. We also simulate several potential threats. Our experimental results demonstrate that our system yields a significant improvement in detecting intrusions and is robust against various attacks, as compared to existing HIDS collaborative systems. The provided experimental results also demonstrate the improved scalability of our system.

The collaborative HIDS framework is presented in Section II and the management model in Section III. The scalability of our system is discussed in Section IV and its robustness against common threats in Section V. Section VI provides experimental evidence of the efficiency, robustness and scalability of our model. Section VII surveys related work and Section VIII summarizes our contributions and future work.

II. HIDS Collaboration Framework

The purpose of this framework is to connect individual HIDSes so that they can securely communicate and cooperate with each other to achieve better detection accuracy. Collaboration is ensured by trust-based cooperation and peer-to-peer communication. The trust management model allows a HIDS to evaluate the trustworthiness of its neighbors based on its own experience with them. The peer-to-peer component provides network organization, management and communication between HIDSes. The collaboration consists of the following two processes.

A. Network Join Process

Before joining the network, a HIDS needs to register to a trusted digital certificate authority and get a public and private key pair which uniquely identifies it. Note that we identify the (machine, user) tuple. This is because a different machine means a different HIDS instance. In addition, a different user of the same machine may have a different configuration of its HIDS. After a peer joins the IDN, it is provided with a preliminary *acquaintance list*. This list is customizable and contains identities (or public keys) of other peers within the network along with their *trust* values. It serves as the contact list for collaboration.

B. Test Messages

Each peer sends out either *requests* for alert ranking, or *test messages*. A test message is a consultation request sent with the intention to evaluate the trustworthiness of another peer in the acquaintance list. It is sent out in a way that makes it difficult to distinguish from a real alert ranking request. The testing peer knows beforehand the *severity* of the alert and uses the received feedback to derive a trust value for the tested peer. This technique helps discover inexperienced or malicious peers within the collaborative network.

III. Trust Management Model

Trust modeling is an important element in an IDN. In this section, we propose a robust and scalable trust model which uses a Bayesian approach to evaluate the trustworthiness between each pair of HIDSes . Specifically, we use a Dirichlet family of probability density functions to estimate the future behavior of a HIDS based on its past history.

A. Satisfaction Mapping

In our model, a HIDS peer sends requests to peer HIDSes and evaluates the satisfaction level of received feedback. Note that the request can be a test message or a real request. The true answer of a test message is known beforehand while that of a real request is estimated after sometime through the observed impact of the corresponding alert.

HIDSes may have different metrics to rank alerts. Snort for example uses three levels (low, medium, high), while Bro allows up to 100 different levels. We assume the existence of a function H, which maps a HIDS alert ranking onto the [0,1] interval where 0 denotes benign traffic and 1 highly dangerous intrusions. H preserves the "more severe than" partial order relationship. That is, if alert a_j is more severe than alert a_i then H preserves that relationship by having $H(a_j) > H(a_i)$.

The satisfaction level of feedback is determined by three factors: the expected answer ($r \in [0,1]$), the received answer ($a \in [0,1]$) and the difficulty level of the test message ($d \in [0,1]$). The larger d is the more difficult it is to correctly answer the request. We use a function $Sat(r,a,d)$ ($\in [0,1]$) to represent the satisfaction of the received answer based on its distance to the expected answer and the difficulty level of the test message, as follows:

$$Sat(r,a,d) = \begin{cases} 1 - \left(\frac{a-r}{\max(c_1 r, 1-r)}\right)^{d/c_2} & a > r \\ \\ 1 - \left(\frac{c_1(r-a)}{\max(c_1 r, 1-r)}\right)^{d/c_2} & a \leq r \end{cases} \quad (1)$$

where c_1 controls the extent of penalty for wrong estimates. It is set > 1 to reflect that estimates lower than the exact answer get stronger penalty than those that are higher. Parameter $c_2 \in R^+$ controls satisfaction sensitivity, with larger values reflecting more sensitivity to the distance between the correct and received answers. The equation also ensures that low difficulty level tests are more severe in their penalty to incorrect answers.

B. Dirichlet-based Model

In our previous work [4], we used a linear model with a forgetting factor to calculate the average satisfaction levels of past interactions. However, this approach does not capture trust modeling uncertainties or provide statistical confidence information on intrusion decisions.

Bayesian statistics provides 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 HIDS 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 [12]. For multi-valued satisfaction levels, Dirichlet distributions are more appropriate.

A Dirichlet distribution [9] 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 HIDS. 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:

$$f(\vec{p}|\xi) = Dir(\vec{p}|\vec{\gamma}) = \frac{\Gamma(\sum_{i=1}^{k}\gamma_i)}{\prod_{i=1}^{k}\Gamma(\gamma_i)} \prod_{i=1}^{k} {p_i}^{\gamma_i - 1} \quad (2)$$

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

Let

$$\gamma_0 = \sum_{i=1}^{k} \gamma_i \quad (3)$$

The expected value of the probability of X to be x_i given the history of observations $\vec{\gamma}$ is given by:

$$E(p_i|\vec{\gamma}) = \frac{\gamma_i}{\gamma_0} \quad (4)$$

In 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:

$$\vec{\gamma}^{(n)} = \sum_{i=1}^{n} \lambda^{t_i} \times \vec{S^i} + c_0 \lambda^{t_0} \vec{S^0} \quad (5)$$

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 $\Delta t_i = t_i - t_{i+1}$. For the purpose of scalability, the $\vec{\gamma}^{(n)}$ in Equation 5 can be rewritten in terms of $\vec{\gamma}^{(n-1)}$, $\vec{S^n}$ and Δt_n as follows:

$$\vec{\gamma}^{(n)} = \begin{cases} c_0 \vec{S^0} & n = 0 \\ \lambda^{\Delta t_n} \times \vec{\gamma}^{(n-1)} + \vec{S^n} & n > 0 \end{cases} \tag{6}$$

C. Evaluating the Trustworthiness of a Peer

After a peer receives the feedback for an alert evaluation, it assigns a satisfaction value to the feedback according to Equation 1. 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 2. Let Y^{uv} be the random variable denoting the weighted average of the probability of each satisfaction level in $\vec{p^{uv}}$.

$$Y^{uv} = \sum_{i=1}^{k} p_i^{uv} w_i \tag{7}$$

In this paper, we adopt a linear pondering factor for the weights $w_i = x_i$. The *trustworthiness* of peer v as noticed by peer u is then calculated as:

$$T^{uv} = E[Y^{uv}] = \sum_{i=1}^{k} w_i E[p_i^{uv}] = \frac{1}{\gamma_0^{uv}} \sum_{i=1}^{k} w_i \gamma_i^{uv} \tag{8}$$

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):

$$\sigma^2[Y] = \sum_{i=1}^{k} \sum_{j=1}^{k} w_i w_j cov[p_i, p_j] \tag{9}$$

Knowing that the covariance of p_i and p_j is given by:

$$cov(p_i, p_j) = \frac{-\gamma_i \gamma_j}{\gamma_0^2(\gamma_0 + 1)} \tag{10}$$

We get:

$$\begin{aligned} \sigma^2[Y] &= \sum_{i=1}^{k} w_i^2 \sigma^2[p_i] + 2 \sum_{i=1}^{k} \sum_{j=i+1}^{k} w_i w_j cov[p_i, p_j] \\ &= \sum_{i=1}^{k} w_i^2 \frac{\gamma_i(\gamma_0 - \gamma_i)}{\gamma_0^2(\gamma_0 + 1)} + 2 \sum_{i=1}^{k} \sum_{j=i+1}^{k} w_i w_j \frac{-\gamma_i \gamma_j}{\gamma_0^2(\gamma_0 + 1)} \\ &= \frac{1}{\gamma_0^3 + \gamma_0^2} \sum_{i=1}^{k} w_i \gamma_i \left(w_i(\gamma_0 - \gamma_i) - 2 \sum_{j=i+1}^{k} w_j \gamma_j \right) \end{aligned} \tag{11}$$

Let $C^{uv} \in (0,1)$ be the confidence level for the value of T^{uv}, we describe it as:

$$C^{uv} = 1 - 4\,\sigma[Y^{uv}] \tag{12}$$

where $4\,\sigma[Y^{uv}]$ is roughly the 95% confidence interval.

D. Feedback Aggregation

Based on their estimated trustworthiness, each peer requests alert consulting only from those peers in its acquaintance list whose trust values are greater than a threshold. After receiving feedback from its acquaintances, a peer *u* aggregates the feedback using a *weighted majority* method as follows:

$$\overline{a}_i^u = \frac{\sum_{T^{uv} \geq th^u, v \in A^u} T^{uv} D^{uv} a_i^{uv}}{\sum_{T^{uv} \geq th^u, v \in A^u} T^{uv} D^{uv}}, \tag{13}$$

where $\overline{a}_i^u$ is the aggregated ranking of alert i from the feedback provided by each peer belonging to the acquaintance list A^u of peer u; $D^{uv} \in [0,1]$ is the proximity weight of peer v with respect to u. th^u is the trust threshold set by u; $a_i^{uv} \in [0,1]$ is the feedback ranking of alert i from v to u.

We introduce *proximity* as a measure of the distance between two peers. In this paper, we consider it to be the geographical distance. This is because HIDSes that are located within the same or close by geographical region are more likely to experience similar intrusions [1] and thus can help each other by broadcasting warnings of active threats. Feedback from nearby acquaintances is therefore more relevant than that from distant ones.

IV. Scalability of Our System

Each HIDS u in our system maintains an acquaintance list with a maximum size N^u. This number can be fixed or slightly updated with the changes in IDN size. However, it is always set to a value small enough to account for scalability. Equation 6 ensures that the process of updating the trustworthiness of a peer after the reception of a response is performed with only three operations, making it linear with respect to the number of answers.

There is a trade-off to be resolved in order to account for scalability in the number of messages exchanged in the IDN. On one hand, the forgetting factor in Equation 6 decays the importance given to existing highly trusted peers. This implies that their corresponding test messages rates need to be above a certain minimal rate. On the other hand, sending too many requests to other peers may compromise scalability. To solve this issue, we adapt the rate of test messages to a given peer according to its estimated trustworthiness. The adaptation policy is provided in Table I, where acquaintances are categorized into highly trustworthy, trustworthy, untrustworthy, and highly untrustworthy. There are three levels of test message rates: $R_l < R_m < R_h$. For the purpose of exploration, acquaintances that are highly untrustworthy are periodically replaced by randomly chosen new peers. We can

TABLE I
ACQUAINTANCE CATEGORIZATION

Peer category	Criterion	Rate
Highly Trustworthy	$0 < th \leq E[Y] - 2\sigma[Y]$	R_l
Trustworthy	$E[Y] - 2\sigma[Y] < th \leq E[Y]$	R_h
Untrustworthy	$E[Y] < th \leq E[Y] + 2\sigma[Y]$	R_m
Highly Untrustworthy	$E[Y] + 2\sigma[Y] < th \leq 1$	R_l

observe that the test message rate to highly trustworthy or highly untrustworthy peers is low. This is because we are confident about our decision of including or not their feedback into the aggregation. A higher test message rate is assigned to trustworthy or untrustworthy peers because their trust values are close to the threshold and hence need to be kept under close surveillance.

Each peer in the system needs to actively respond to others' requests in order to keep up its trustworthiness and be able to receive prompt help when needed. However, actively responding to every other peer will cause bottleneck situations. Therefore, as a consultant to others, a peer would like to limit the rate of answers it provides. In this regard, each peer in our system would respond to requests with a priority proportional to the amount of trust it places on the source of the request. It will give higher priority to highly trusted friends. This obeys the social norm: "Be nice to others who are nice to you", and also provides incentives for encouraging peers to act honestly in order to receive prompt help in times of need.

V. ROBUSTNESS AGAINST COMMON THREATS

Trust management can effectively improve network collaboration and detect malicious peers. However, the trust management itself may become the target of attacks and be compromised. In this section, we describe common attacks 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) [2]. This malicious peer uses fake identities to gain larger influence over the false alert ranking on others in the network. Our defense against sybil attacks relies on the design of the authentication mechanism. Authentication makes registering fake identities difficult. In our model, the certificate issuing authority only allows one identity per (user, machine) tuple. In addition, our trust management model requires HIDSes to first build up their trust before they can affect the decision of others, which is costly to do with many fake identities. Thus, our security and trust mechanisms protect our collaborative network from sybil attacks.

2) Newcomer attacks: occur when a malicious peer can easily register as a new user [8]. 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 assigning low trust values to all newcomers, so their feedback on the alerts is simply not considered by other peers during the aggregation process.

TABLE II
SIMULATIONS PARAMETERS

Parameter	Value	Description
R_l	2/day	Low test message rate
R_m	10/day	Medium test message rate
R_h	20/day	High test message rate
λ	0.9	Forgetting factor
th	0.8	Trust threshold for aggregation
c_0	10	Priori Constant
c_1	1.5	Cost rate of low estimate to high estimate
c_2	1	Satisfaction sensitivity factor
s	4	Size of grid region
k	10	Number of satisfaction levels

3) Betrayal attacks: occur when a trusted peer suddenly turns into a malicious one and starts sending false alerts or even malware. A trust management system can be degraded dramatically because of this type of attacks. We employ a mechanism which is inspired by the social norm: "It takes a long-time interaction and consistent good behavior to build up a high trust, while only a few bad actions to ruin it." When a trustworthy peer acts dishonestly, the forgetting factor (Equation 6) causes its trust value to drop down quickly, hence making it difficult for this peer to deceive others or gain back its previous trust within a short time.

4) Collusion attacks: happen when a group of malicious peers cooperate together by providing false alert rankings in order to compromise the network. In our system, peers will not be adversely affected by collusion attacks. In our trust model each peer relies on its own knowledge to detect dishonest peers. In addition, we use test messages to uncover malicious peers. Since the test messages are sent in a random manner, it will be difficult for malicious peers to distinguish them from actual requests.

5) Inconsistency attacks: happen when a mailcious peer repeatedly changes its behavior from honest to dishonest in order to degrade the efficiency of the IDN. Inconsistency attacks are harder to succeed in the Dirichlet-based model because of the use of the forgetting factor and the dynamic test message rate, which makes trust values easy to lose and hard to gain. This ensures that the trust values of peers with inconsistant behaviour remain low and hence have little impact.

VI. SIMULATIONS AND EXPERIMENTAL RESULTS

In this section, we present a set of experiments used to evaluate the effeciency, scalability and robustness of our trust management model in comparison with existing ones [4][3]. Each experimental result presented in this section is derived from the average of a large number of replications with an overal negligible confidence interval.

A. Simulation Setting

The simulation environment uses an IDN of n HIDS peers randomly distributed over an $s \times s$ grid region. The proximity distance is given by the minimum number of square steps between each two peers. The expertise level of a peer can

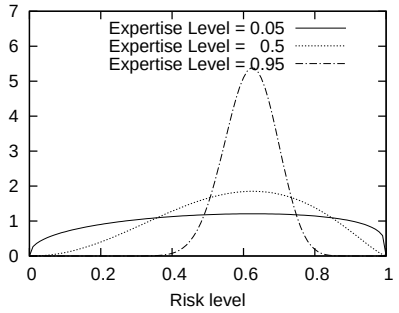

Fig. 1. Decision Density Function for Expertise Levels

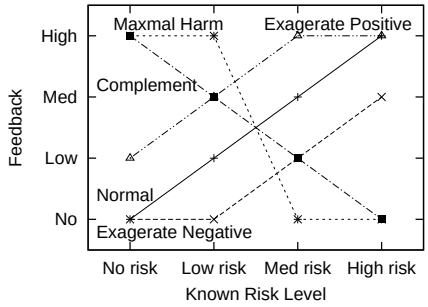

Fig. 2. Feedback Curves for Different Deception Strategies

be low (5%), medium (50%) or high (95%). In the beginning, each peer builds an initial acquaintance list based on the communication cost (proximity). The initial trust value of every peer in the acquaintance list is 50%. To test the trustworthiness of acquaintances, each peer sends out test messages following a Poisson process with rates according to Table I. The parameters we used are shown in Table II.

B. Modeling the Expertise Level of a Peer

To reflect the expertise level of each peer, we use a Beta distribution to simulate the decision model of answering requests. A Beta density function is given by:

$$\begin{aligned} f(p|\alpha,\beta) &= \frac{1}{B(\alpha,\beta)}p^{\alpha-1}(1-p)^{\beta-1} \\ B(\alpha,\beta) &= \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}dt \end{aligned} \quad (14)$$

where $f(p|\alpha,\beta)$ is the probability that a peer with expertise level l answers with a value of $p \in [0,1]$ to an alert of difficulty level $d \in [0,1]$. Higher values for d are associated to attacks that are difficult to detect, i.e. many peers fail to identify them. Higher values of l imply a higher probability of producing correct alert rankings.

Let r be the expected ranking of an alert. We define α and β as follows:

$$\begin{aligned} \alpha &= 1 + \frac{l(1-d)}{d(1-l)}\sqrt{\frac{r}{1-r}}\sqrt{\frac{2}{l}-1} \\ \beta &= 1 + \frac{l(1-d)}{d(1-l)}\sqrt{\frac{1-r}{r}}\sqrt{\frac{2}{l}-1} \end{aligned} \quad (15)$$

For a fixed difficulty level, the above model has the property of assigning higher probabilities of producing correct rankings to peers with higher levels of expertise. A peer with expertise level l has a lower probability of producing correct rankings for alerts of higher difficulty ($d > l$). $l = 1$ or $d = 0$ represent the extreme cases where the peer can always accurately rank the alert. This is reflected in the Beta distribution by $\alpha, \beta \to \infty$. Figure 1 shows the feedback probability distribution for peers with different expertise levels, where we fix the expected risk level to 0.7 and the difficulty level of test messages to 0.5.

C. Deception Models

We model four deception models for a dishonest peer: *complementary*, *exaggerate positive*, *exaggerate negative*, and *maximal harm*. The first three deception models are described in [11], where an adversary may choose to send feedback about the risk level of an alert that is respectively opposite to, higher, or lower than the true risk level. We propose a maximal harm model where an adversary always chooses to report false feedback with the intention to bring the most negative impact to the request sender. Figure 2 shows the feedback curve for the different deception strategies. For instance, when a deceptive peer using the maximal harm strategy receives a ranking request and detects that the risk level of the request is "medium", it sends feedback "no risk" because this feedback can maximally deviate the aggregated result at the sender side.

D. Trust Values and Confidence Levels for Honest Peers

The first experiment studies the effectiveness of the collaboration and the importance of our trust management. In this experiment, all peers are honest. We simulate the scenario where each peer u has a fixed size N^u of its acquaintance list. The peers are divided into three equally-sized groups of *low*, *medium* and *high* expertise levels respectively. The first phase of the simulation is a learning period (50 days), during which peers learn about each other's expertise level by sending out test messages. Figure 3 shows the resulting average trust values of the 30 acquaintances of peer u. 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.

To study the impact of different test message rates on the confidence level of trust estimation (Equation 12), we conduct a second experiment to let u use a fixed test message rate in every simulation round. The rate of sending test messages starts with one message per day and increases by five for every simulation round. We plot the confidence level of trust evaluation for each test message rate in Figure 4. We can observe that the confidence level increases with the increase of the test message rate. This confirms our argument that sending more test messages improves the confidence of trust estimation. We also observe that the confidence levels increase with the expertise levels. This is because peers with higher

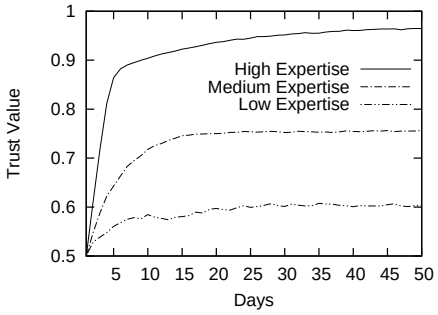

Fig. 3. Convergence of Trust Values for Different Expertise Levels

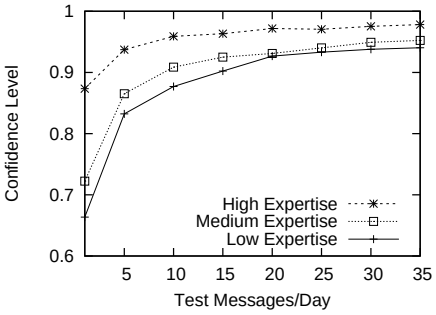

Fig. 4. Confidence Levels of Estimation for Different Test Message Rates

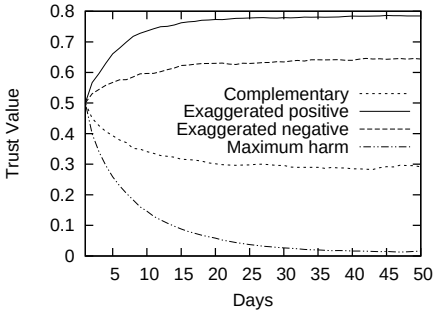

Fig. 5. Trust Values of Deceptive Peers with Different Deception Strategies

expertise levels tend to perform more consistently.

E. Trust Values for Dishonest Peers

The purpose of this experiment is to study the impact of dishonest peers using the four different deception strategies described in Section VI-C. To study the maximum impact of these deception strategies, we only use peers with a *high* expertise level as deceptive adversaries since they are more likely to know the true answers and can perform the deception strategies more accurately.

In this experiment, we let peer u have an acquaintance list of 40 dishonest peers divided into four groups. Each group uses one of the four deception models: complimentary, exaggerate positive, exaggerate negative, and maximal harm. We use a dynamic test message rate and observe the convergence curve of the average trust value for each group of deceptive peers. Results are plotted in Figure 5.

We notice that the trust values of all adversary peers converge to stable values after 30 days of the learning phase. It is not surprising that adversary peers using the maximal harm strategy have the lowest trust values, while adversary peers using the complimentary strategy have the second lowest ones. The converged trust values of adversary peers using exaggerate positives are higher than those using exaggerate negatives. This is because we use an asymmetric penalization mechanism for inaccurate replies ($c_1 > 1$ in Equation 1). We penalize more heavily peers that untruthfully report lower risks than those which untruthfully report higher risks.

F. Robustness of Our Trust Model

The goal of this experiment is to study the robustness of our trust model against various insider attacks. For the newcomer attack, malicious peers white-wash their bad history and re-register as new users to the system. If the trust value of a newcomer can increase quickly based on its short term good behavior, the system is then vulnerable to newcomer attacks. However, a newcomer attack is difficult to succeed in our model. In our model, we use parameter c_0 in Equation 6 to control the trust value increasing rate. When c_0 is larger, it takes longer for a newcomer to gain a trust value above the trust threshold.

We compare our Dirichlet-based model with our previous model [4] and the model of Duma et al. [3] in Figure 6. We observe that in the Duma et al. model, the trust values of new users increase very fast and reach the aggregation trust threshold (80%) in the first day, which reveals a high vulnerability to newcomer attacks. The reason for this is that their model does not assign an initial trust to new peers and therefore their trust values change very fast in the beginning. In the model we developed in [4], the trust values increase in a slower manner and reach the trust threshold after three days. However, that model is not flexible in that it does not offer control over the trust increase speed. In the Dirichlet-based model, the trust increase speed is controlled by the priori constant c_0. For $c_0 = 10$, it takes a newcomer four to five days of consistent good behavior to reach the same trust value. Larger values of c_0 make it even slower to reach high trust, hence offering robustness against newcomer attacks.

The second possible threat is the betrayal attack, where a malicious peer first gain a high trust value and then suddenly starts to act dishonestly. This scenario can happen, for example, when a peer is compromised. To demonstrate the robustness of our model against this attack type, we set up a scenario where u has seven peers in its acquaintance list, of which six are honest with an expertise evenly divided between low, medium, and high. The malicious one has high expertise and behaves honestly in the first 50 days. After that, it launches a betrayal attack by adopting a maximal harm deceptive strategy. We observe the trust value of the betraying peer and the satisfaction levels of aggregated feedback in each day with respect to u.

Figure 7 shows the trust value of the betraying peer before and after the launching of the betrayal attack when respectively using Duma et al., our previous and our new trust models. For the Duma et al. model, the trust value of the malicious peer slowly drops after the betrayal attack. This is because their model does not use a forgetting factor, hence providing the previous honest behavior of a malicious peer with a heavy impact on the trust calculation for a considerable amount of time. The trust value of the betraying peer drops much faster using our previous model, while the fastest rate is observed when using our Dirichlet-based model. This is because both

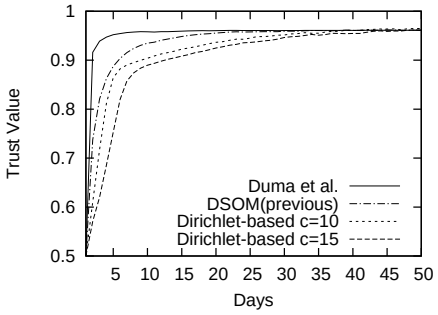

Fig. 6. Trust Values of Newcomers under Different Trust Models

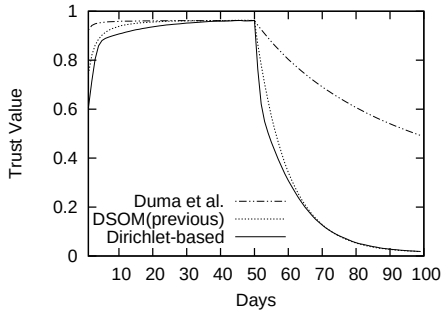

Fig. 7. Trust of Malicious Peers under Betrayal Attack

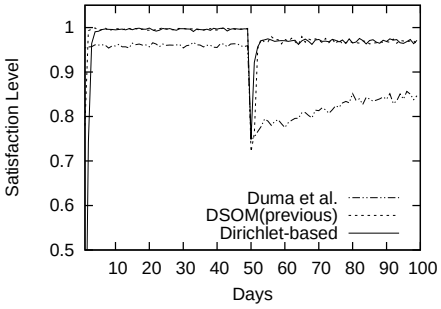

Fig. 8. Impact on Accuracy of Betrayal Attack

models use a forgetting factor to pay more attention to the more recent behavior of peers.

We also notice that the Dirichlet-based model has a slight improvement over our previous model. The Dirichlet-based model adopts the dynamic test message rate and can react more swiftly. The rate of sending messages to malicious peers increases as soon as they start to behave dishonestly. Higher rates of test messages help in the prompt detection of dishonest behavior. However, in our previous model, the test message rate remains the same. This phenomenon can be further observed in Figure 9.

The results for the satisfaction levels of aggregated feedback with respect to u before and after the betrayal attack are shown in Figure 8. We notice that the satisfaction level of u for the aggregated feedback drops down drastically in the first day following the learning period and recovers after that in all three models. The recovery period is however much shorter for the Dirichlet-based and our previous models. This is again attributed to the use of the forgetting factor. The Dirichlet-based model has a slight improvement in the recovering speed over our previous model. This is because in the Dirichlet-based model, the trust values of betraying peers drop under the aggregation threshold faster than our previous model. Therefore, the impact of betraying peers is eliminated earlier than in the previous model.

G. Scalability of Our Trust Model

The result of test message rates under betrayal attack is shown in Figure 9. We notice that in our Dirichlet-based model, the average test message rates for highly trusted as well as highly non trusted peers are the lowest. The average test message sending rate to peers with the medium expertise level is higher but still below the medium rate (R_m). Compared to our previous model, the average message sending rate is much lower, which demonstrates the improved scalability of our Dirichlet-based model. Note that the spike from the betraying group on around day 50 is caused by the drastic increment of the test message rate. The sudden change of a highly trusted peer behavior will cause the trust confidence level to drop down quickly. The rate of sending messages to this peer then switches to R_h accordingly.

H. Efficiency of Our Trust Model

To demonstrate the efficiency of our Dirichlet-based trust model, we conduct another experiment to evaluate the intrusion detection accuracy. In this experiment, we let peer u have 15 acquaintances, which are evenly divided into low, medium, and high expertise groups. Among the expert peers, some are malicious and launch inconsistency attacks synchronously to degrade the efficiency of the IDN. More specifically, in each round of behavior changing, these malicious peers adopt the maximal harm deception strategy for two days followed by six days of honest behavior.

In Figure 10, we vary the percentages of malicious peers from 0% to 80%. We inject daily intrusions to peer u with medium difficulty (0.5) and random risk levels. We then plot the average satisfaction level for the aggregated feedback. We observe that our Dirichlet-based model outperforms the others. This is because the dynamic test message rate in Dirichlet-based model causes the trust of malicious peers to drop faster and increase slower, hence minimizing the impact of dishonest behavior. Among the three models, Duma et al. has the least satisfaction level because of its slow response to sudden changes in peer behavior and its aggregation of all feedback from even untrustworthy peers.

Figure 11 shows the success rate of peer u in detecting intrusions. We notice that both our previous model and the Duma et al. model cannot effectively detect intrusions when the majority of peers are malicious. Our Dirichlet-based model shows excellent efficiency in intrusion detection even in the situation of a dishonest majority.

VII. Related Work

Most of the existing work on distributed collaborative intrusion detection relies on the assumption that all peer HIDSes are trusted and faithfully report intrusion events [5][7]. These systems can be easily compromised if some of the peers are (or become) dishonest. Duma et al. propose in [3] a trust management model to identify dishonest insiders and a trust-aware collaboration mechanism for correlating intrusion alerts. Their trust management scheme uses the past experience of each peer to predict the trustworthiness of other peers. However, their trust model does not address security issues

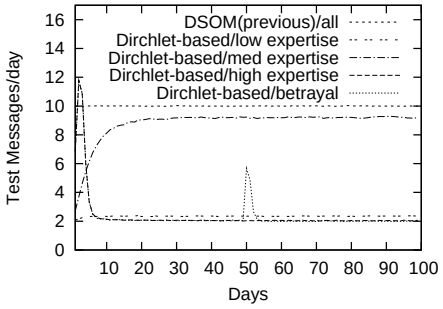

Fig. 9. Comparison of Average Test Message Rates under Different Models

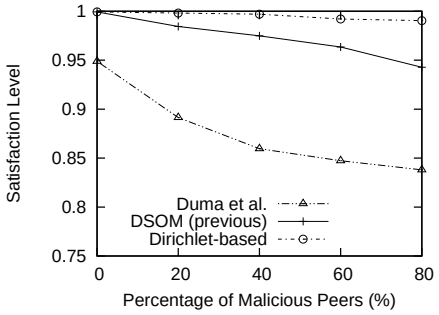

Fig. 10. Aggregated Feedback under Inconsistency Attack

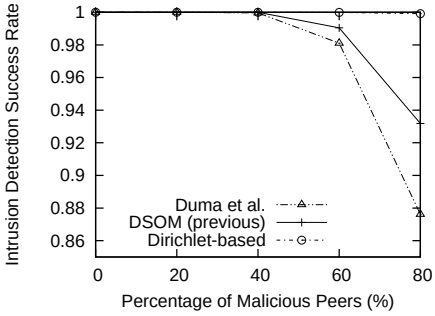

Fig. 11. Intrusion Detection Success Rate under Inconsistency Attack

within the collaborative network. For instance, in their system, the past experience of a peer has the same impact on its final trust value regardless of the age of its experience, therefore making it vulnerable to newcomer and betrayal attacks. In our model, we use a forgetting factor when computing the trust to put more emphasis on the recent experience of the peer. The Duma et al. model integrates feedback from all the peers in the IDN, which does not scale. Our model avoids this deficiency by only integrating feedback from trusted peers.

Different models have been proposed for trust management in distributed networks. Jiang and Baras [6] use a global reputation management to evaluate distributed trust by aggregating votes from all peers in the network. Sun et al. [10] propose an entropy-based model and a probability-based model, which are used to calculate the indirect trust, propagation trust and multi-path trust. These models have a lot of overhead and are not suitable for our system because the peers can be easily compromised. They also suffer from collusion attacks since their trust values are based on the votes from others.

Our model is also distinguished from the trust models developed for the application of e-marketplaces [12]. We introduce the concepts of expertise level and proximity to improve the accuracy of intrusion detection. We also allow the peer HIDSes to send test messages to establish better trust relationships with others. The alert risk ranking is categorized into multiple levels as well.

Our previous work [4] propose a robust trust management model that uses test messages to gain personal experience and a forgetting factor to emphasize most recent experiences. However, this model needs to repeatedly aggregate all past experience with a peer when updating its trust, which makes it not scalable over time. It also lacks a sound theoretical basis. Our new model uses Dirichlet distributions to model peer trust. It limits the size of the acquaintance list and uses dynamic test message rates in order to account for better scalability. Also, the dynamic adaptation in test message rates provides improved robustness over our previous model.

VIII. Conclusions and Future Work

In this paper, we present an efficient trust-based HIDS IDN management solution, which is robust against common insider threats and offers strong scalability properties. The experimental results demonstrate the improved performance of our model in detecting intrusions, as well as its robustness and scalability.

Our work contributes to the area of trust-based collaborative intrusion detection and achieves the important properties of efficiency, robustness and scalability in IDN management.

As future work, we plan to develop and deploy a real IDN using existing intrusion detection systems. We will also investigate more sophisticated types of insider threats, such as collusion attacks. Furthermore, we will design effective incentive approaches so as to avoid free-rider problems and offer better rewards to honest participants.

Acknowledgment

The authors would like to thank Quanyan Zhu for his participation in the discussion on the trust model formulation and his contribution to the editing of this paper.

References

[1] J. Aycock. Painting the internet: A different kind of warhol worm. *Technical Report, TR2006-834-27, University of Calgary*, 2006.

[2] J. Douceur. The sybil attack. *Peer-To-Peer Systems: First International Workshop, IPTPS 2002, Cambridge, MA, USA, March 7-8, 2002*, 2002.

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

[4] 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.

[5] 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.

[6] T. Jiang and J. Baras. Trust evaluation in anarchy: A case study on autonomous networks. In *INFOCOM*. IEEE, 2006.

[7] Z. Li, Y. Chen, and A. Beach. Towards scalable and robust distributed intrusion alert fusion with good load balancing. In *LSAD '06*, 2006.

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

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

[10] Y. Sun, Z. Han, W. Yu, and K. Liu. A trust evaluation framework in distributed networks: Vulnerability analysis and defense against attacks. In *INFOCOM*. IEEE, 2006.

[11] 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.

[12] 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. ACM.