Design of Intrusion Sensitivity-Based Trust Management Model for Collaborative Intrusion Detection Networks

Wenjuan Li, Weizhi Meng, Lam-For Kwok

Design of Intrusion Sensitivity-Based Trust Management Model for Collaborative Intrusion Detection Networks

Wenjuan Li, Weizhi Meng* and Lam-For Kwok

Department of Computer Science, City University of Hong Kong, Hong Kong, China
wenjuan.li@cityu.edu.hk,yuxin.meng@cityu.edu.hk

Abstract. Network intrusions are becoming more and more sophisticated to detect. To mitigate this issue, intrusion detection systems (IDSs) have been widely deployed in identifying a variety of attacks and collaborative intrusion detection networks (CIDNs) have been proposed which enables an IDS to collect information and learn experience from other IDSs with the purpose of improving detection accuracy. A CIDN is expected to have more power in detecting attacks such as denial-of-service (DoS) than a single IDS. In real deployment, we notice that each IDS has different levels of sensitivity in detecting different types of intrusions (i.e., based on their own signatures and settings). In this paper, we propose a machine learning-based approach to assign intrusion sensitivity based on expert knowledge and design a trust management model that allows each IDS to evaluate the trustworthiness of others by considering their detection sensitivities. In the evaluation, we explore the performance of our proposed approach under different attack scenarios. The experimental results indicate that by considering the intrusion sensitivity, our trust model can enhance the detection accuracy of malicious nodes as compared to existing similar models.

Keywords: Network Security, Intrusion Detection, Trust Management, Intrusion Sensitivity, Collaborative Intrusion Detection Network.

1 Introduction

Network intrusions (e.g., worms, spamware, Trojans, virus, etc.) have become more and more sophisticated and harmful [25]. To mitigate this problem, intrusion detection systems (IDSs) have been widely deployed in current computers and networks aiming to defend against a variety of attacks, and these detection systems have already become an essential component for current defense mechanism [21].

Traditionally, these intrusion detection systems can be classified into two general types based on their protected environments[1] [21]: *host-based IDS (HIDS)*

* Corresponding author and is previously known as Yuxin Meng.

[1] Based on the detection approaches, these intrusion detection systems can be roughly classified as signature-based IDS [27] and anomaly-based IDS [7].

and *network-based IDS (NIDS)*. The HIDS detects abnormal executions by logging and analyzing system events within a single host while the NIDS is mainly monitoring and analyzing network traffic for identifying suspicious activities. But in a large-scale network environment, a single IDS cannot detect some certain attacks such as denial-of-service (DoS) and distributed DoS (DDoS). The potential damage of these attacks can be significant if failed detected (i.e., causing paralysis of the entire network). In addition, an isolated IDS would be easily bypassed by unknown or novel exploits.

To resolve this issue, IDS collaboration is an effective way to enhance the detection capability of a single IDS. Thus, intrusion detection network (IDN) has been developed, which is a collaborative IDS network, with the purpose of strengthening a single IDS by collecting knowledge and learning experience from other IDS nodes. This collaborative IDN (CIDN) [28] is expected to enhance the overall detection accuracy of intrusion assessment and improve the possibility of identifying novel attacks. However, attackers can compromise some peers (or *some IDS nodes*) in the CIDN and utilize these compromised peers to invade or against the collaborative network. These malicious peers can make use of some attacks including Sybil attacks, newcomer attacks, betrayal attacks to lower the effectiveness and efficiency of a CIDN by sending false information and compromising other honest IDS nodes within the network. In these cases, designing a robust CIDN (i.e., effectively evaluating the trustworthiness of each IDS in the network) becomes very crucial and essential to improve its detection capability and protect this network against insider attacks.

Contributions. In our previous work [12], we have identified that each IDS has different levels of sensitivity in detecting particular intrusions and proposed a notion of *intrusion sensitivity*. Our goal of this paper is thus designing a trust management model based on *intrusion sensitivity* to improve the robustness of CIDNs. In particular, we begin by reviewing recent works of building trust models regarding intrusion detection. We then detail the notion of *intrusion sensitivity* and build an *intrusion sensitivity-based trust management model* for a CIDN. Our contributions of this work can be summarized as below:

- We review some related works about establishing trust models in the field of intrusion detection and introduce the tuned CIDN's framework to adapt to our model, which consists of several major components including IDS nodes, trust management component, collaboration component, communication component and query component.
- Our previous work [12] proposed a notion of *intrusion sensitivity* that measures the detection sensitivity of an IDS in detecting different kinds of intrusions. This work we thus aim to develop an *intrusion sensitivity-based trust management model* for CIDNs. To automatically realize the assessment of *intrusion sensitivity*, we further develop a *query component* and an expert knowledge-based KNN classifier to allocate the sensitivity level.
- In the evaluation, we simulated a collaborative intrusion detection network and certain attacks to investigate the performance of our proposed trust management model under different attack scenarios. The experimental re-

sults indicate that our proposed model by considering the *intrusion sensitivity* is more efficient and sensitive in detecting malicious nodes as compared to other similar trust models.

The remaining parts of this paper are organized as follows. In Section 2, we review some related works about trust models in collaborative intrusion detection networks; Section 3 describes our proposed trust model in detail including CIDN framework, intrusion sensitivity and trust evaluation, and analyzes the robustness of the trust model against several common attacks. Section 4 presents experimental settings and describes experimental results and Section 5 analyzes some limitations and challenges. Finally, we conclude our work with future directions in Section 6.

2 Related Work

Intuitively, an isolated (or single) intrusion detection system has no information about the whole protected environment and thus is more likely to be bypassed by novel intrusions. To resolve this issue, collaborative intrusion detection networks (CIDNs) [28] have been proposed and implemented which enable an IDS node to achieve more accurate detection by collecting and learning useful information from other IDS nodes.

A number of trust models have been proposed for CIDNs. For instance, Janakiraman and Zhang [9] proposed *Indra*, a distributed scheme based on sharing information between trusted peers in a network to guard a peer-to-peer network as a whole against intrusion attempts. Li *et al.* [11] identified that most distributed intrusion detection systems (DIDS) relied on centralized fusion, or distributed fusion with unscalable communication mechanisms, and then proposed a DIDS based on the emerging decentralized location and routing infrastructure. The experimental results showed that their methods could greatly outperform the traditional hierarchical approach when facing large amounts of diverse intrusion alerts. However, these approaches assume that all peers are trusted which is vulnerable to insider attacks (i.e., some nodes become malicious). Several distributed intrusion detection systems can be classified as:

- *Centralized/Hierarchical systems*: Emerald [16] and DIDS [22];
- *Publish/subscribe systems*: COSSACK [15] and DOMINO [29];
- *P2P Querying based systems*: Netbait [2] and PIER [8].

To identify insider attacks, Duma *et al.* [3] proposed a P2P-based overlay for intrusion detection (Overlay IDS) that mitigated the insider threat by using a trust-aware engine for correlating alerts and an adaptive scheme for managing trust. The trust-aware correlation engine is capable of filtering out warnings sent by untrusted or low quality peers, while the adaptive trust management scheme uses past experiences of peers to predict their trustworthiness. But a major issue is that the past experience of a peer has the same impact regardless of the age

of its experience. To resolve this problem, Fung *et al.* [4] proposed a HIDS collaboration framework that enables each HIDS to evaluate the trustworthiness of others based on its own experience by means of a forgetting factor. The forgetting factor can give more emphasis on the recent experience of the peer. Later, Fung *et al.* [5] improved their proposed trust management model by using a Dirichlet-based model to measure the level of trustworthiness among IDS nodes according to their mutual experience. This model had strong scalability properties and was robust against common insider threats and the experimental results demonstrated that the new model could improve robustness and efficiency. As the mechanism of feedback aggregation is a key component in the above trust model, Fung *et al.* [6] further applied a Bayesian approach to feedback aggregation to minimize the combined costs of missed detection and false alarm. Their experiments indicated that the Bayesian approach could make an improvement in the true positive detection rate and a reduction in the average cost.

In addition, Quercia *et al.* [18] proposed a distributed trust framework that satisfied a broader range of properties, which evolved an expressive and tractable trust calculation based on Bayesian formalization, protected user anonymity and integrated a risk-aware decision module. Then, Li *et al.* [10] proposed an objective trust management framework ($OTMF$) using a modified Bayesian approach where the trust in the provider of second-hand information is considered when evaluating trust. They further conducted a performance evaluation and security analysis on $OTMF$, and the results showed that the $OTMF$ was more effective and robust as compared to similar frameworks.

Many theories have also been investigated to evaluate the trustworthiness of communication entities such as Information Theory, Game theory and Grey Theory. For example, Sun *et al.* [24] presented an information theoretic framework to quantitatively measure trust and model trust propagation in Ad Hoc networks. In their framework, trust is a measure of uncertainty with its value represented by entropy. They developed four Axioms that addressed the basic understanding of trust and the rules for trust propagation. The simulations showed that their approach could significantly improve the network throughput as well as effectively detect malicious behaviors in Ad Hoc networks. Tuan [26] used the game theory to model and analyze the processes of reporting and exclusion in a P2P network. They found that if a reputation system was not incentive compatible, the more numbers of peers in the system, the less likely that anyone will report about a malicious peer. Later, Cai *et al.* [1] proposed a novel risk assessment method based on grey theory to identify the malicious recommendations. They further showed that grey theory was suitable for P2P networks.

In our previous work [12], we identified that different IDSs may have different levels of sensitivity in detecting different types of intrusions and proposed a notion of *intrusion sensitivity*, which helps detect intrusions and correlate IDS alerts through emphasizing the impact of an *expert IDS*.[2] Based on the notion, in this work, we aim to design an *intrusion sensitivity-based trust management*

[2] Note that these IDS nodes are assumed to have more powerful capability and sensitivity in identifying some certain malicious activities.

model for CIDNs and compare our model with some similar models in the evaluation. The experimental results under several attack scenarios indicate that our approach can improve the accuracy of identifying insider attacks as compared to the existing trust models.

3 CIDN Framework and Intrusion Sensitivity-Based Trust Management Model

A CIDN can enable single IDS nodes to connect, communicate and cooperate with others. In this work, we design a *query component* to allocate its values and consequently establish a trust management model. In this section, we modify a CIDN framework (without a centralized server) based on our previous work [12], introduce how to assign the value of *intrusion sensitivity* and how to evaluate the trustworthiness of an IDS node.

3.1 CIDN Design

In Fig. 1, we describe the key components of the adopted CIDN framework: *IDS nodes*, *trust management component*, *query component*, *collaboration component* and *communication component*. This trust model allows an IDS node to evaluate the trustworthiness of others based on its own and others' experience.

IDS Nodes. In the framework, each IDS node (based on either a HIDS or a NIDS) can choose its collaborators according to its own experience. These nodes are associated if they have a collaborative and cooperative relationship. Each node can maintain a list of their collaborated nodes. In this paper, we call this list as *partner list*. The *partner list* is customizable and contains public keys of other nodes and their current trust values.

If a node requests to join this collaborative network, it needs to register to a trusted certificate authority (CA) and get its unique proof of identity (including a public key and a private key). For example as shown in Fig. 1, if node D wants to join the CIDN, then it can send a request to a network node, say node A. After receiving the request, node A can send back the decision (either accept or decline). If node D is accepted to join the network, it can then receive an initial *partner list* from node A.

Trust Management Component. This component is responsible for evaluating the trustworthiness of other nodes. In this work, we mainly consider two types of trust: *feedback-based trust* and *packet-based trust*, aiming to provide a comprehensive trust evaluation in this component:

- *Feedback-based trust* is established based on the feedbacks from partner nodes (which appear in the *partner list*). The feedback will be sent and received by a collaboration component.
- *Packet-based trust* is computed based on the received benign packets and total packets from the target node. This type of trust is objective and is helpful for determining a trusted route and identify malicious nodes.

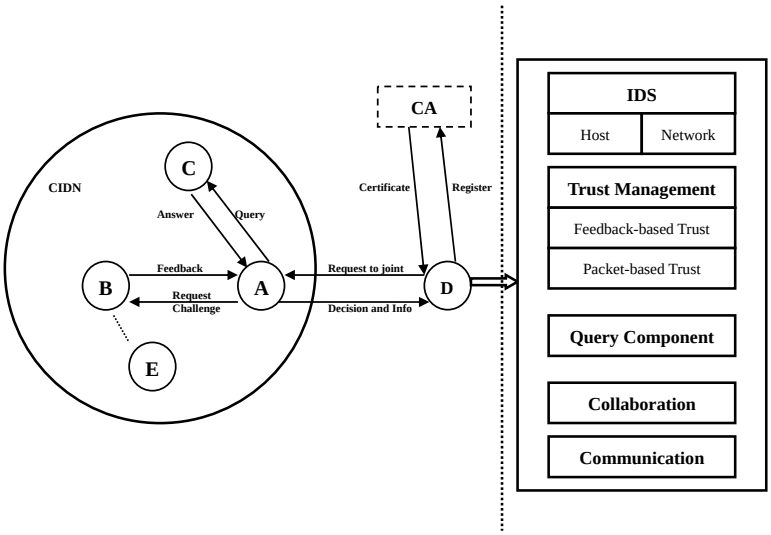

Fig. 1. The framework of our designed collaborative intrusion detection network integrating with a query component (aiming to request *intrusion sensitivity*).

Query Component. This is a key component in our designed framework, which can send a set of *queries* to a target node in which a *query* mainly contains a series of alarms (e.g., 5 to 10) while *answers* are alarm rankings sent back from the target node. Basically, these *answers* are decided by the experience, configuration and settings of each IDS node. For example, Snort [23] has classified its rules to three different priorities, thus, the corresponding triggered alarms can be ranked according to the matched rules. On the contrary, certain alarms cannot be correctly classified if lacking of some rules. In this case, the specific *intrusion sensitivity* of an IDS node can be determined according to the *answers*.

As shown in Fig. 1, if node A sends a *query* to node C, then node C will send back an *answer* to node A. Intuitively, different IDS nods may have different levels of *intrusion sensitivity* with regard to each individual type of intrusions. For example, if an IDS node has more powerful rules in detecting a certain attack like denial of service attack (DoS), then it can send back a more accurate alarm ranking for this attack and can be allocated a higher sensitivity level for this particular attack. Based on the different levels of *intrusion sensitivity*, we can emphasize the impact of *expert nodes* in detecting malicious nodes and attacks. In this work, the levels of *intrusion sensitivity* can be automatically assigned by means of a machine learning classifier (e.g., KNN) after receiving the *answers*. The details will be discussed later.

Collaboration Component. This component is mainly responsible for assisting a node to evaluate the trustworthiness (namely *feedback-based trust*) of others by sending out *requests* and *challenges* (in a period of time), and collecting the corresponding *feedback*.

- *Requests* can be sent by an IDS node for alert consultation. For example, an IDS node may request other nodes to help determine the ranking of several alerts. A *request* is mainly used for alert aggregation and is beyond the scope of this paper.
- *Challenges* are sent by an IDS node for evaluating the trustworthiness of another node in the *partner list*. In particular, this node knows the desirable feedback for the challenges so that it can evaluate the trustworthiness of other nodes by analyzing the received feedback (answers).
- *Feedback* will be sent back from other IDS nodes for the corresponding *requests* and *challenges*. If an IDS node receives a request or challenge, this component will send back its feedback as the answers. As shown in Fig. 1, if node A sends a *request/challenge* to node B, then node B will send back relevant feedback.

Communication component. This component is responsible for connecting with other IDS nodes and providing network organization and communication between IDS nodes. For instance, for a HIDS-based CIDN, this component can use P2P. In addition, this component can assist a node to evaluate the trustworthiness (namely *packet-based trust*) of other nodes by recording the number of transmitted packets and the state of packets (e.g., benign) based on IDS's rules or normal profiles. The details of trust computation will be discussed next.

3.2 Trust Evaluation

To evaluate the trustworthiness of a target node, an IDS node can sent a *challenge* to this target periodically using a random generation process. When receiving the feedback from the target node, the IDS node can give a score to reflect its satisfaction level. As we define two types of trust including *feedback-based trust* (T_{fd}) and *packet-based trust* (T_{pt}), we develop a single metric called *overall trust* (T_{total}) to facilitate the trust evaluation as follows:

$$T_{total} = W_1 \times T_{fd} + W_2 \times T_{pt} \tag{1}$$

where W_1 and W_2 are weight values and $W_1 + W_2 = 1$. For the feedback-based trust $T_{fd}^{i,j}$ of node i according to node j, we can compute it by using the equation described as below:

$$T_{fd}^{i,j} = w_s \frac{\sum_{k=0}^{n} F_k^j \lambda^{tk}}{\sum_{k=0}^{n} \lambda^{tk}} \tag{2}$$

where $F_k^j \in [0, 1]$ is the score of the received feedback k and n is the total number of feedback. λ is a *forgetting factor* that assigns less weight to older feedback response. w_s is a *significant weight* depends on the total number of received feedback, if there is only a few feedback under a certain minimum m, then $w_s = \frac{\sum_{k=0}^{n} \lambda^{tk}}{m}$, and otherwise $w_s = 1$.

On the other hand, in this work, the *packet-based trust* of node i according to node j can be computed based on our another work [14] as below:

$$T_{pt}^{i,j} = \frac{k+1}{N+2} \tag{3}$$

where k is the number of received benign packets and N is the total number of received packets. The detailed derivation and computation can refer to [14].

Assignment of Intrusion Sensitivity. As described above, each IDS node can consult alert ranking from other nodes by sending out *queries*. After receiving the *answers*, a node thus can evaluate the *intrusion sensitivity* of other nodes accordingly. However, to automatically assign the levels of *intrusion sensitivity* is a big challenge [12].

To address this issue, we identify that a machine learning classifier based on expert knowledge can be utilized. In this work, we thus use a k-nearest neighbors algorithm (KNN) to automatically allocate the values of *intrusion sensitivity*. The reasons of selecting this classifier are shown as below:

- The KNN classifier aims to classify objects based on the closest training examples in the feature space. That is, an object is classified in terms of its distances to the nearest cluster. In [13], this classifier has proven to be effective in intrusion detection with a high detection accuracy.
- In addition, this classifier can achieve a faster speed with lower computational burden as compared to other classifiers like neural networks in the phases of both training and classification. These properties are desirable when deployed in a resource-limited platform like an *IDS node*.

To evaluate and assign the *intrusion sensitivity* of other nodes using the KNN classifier, there are generally two steps shown as follows:

- We first obtain several scores for the feedback based on expert knowledge and build a classifier model. In this work, we employ three experts from recognized organization regarding intrusion detection and Honeypot[3] to give scores for different sets of queries and answers. We then use a KNN classifier to establish a model.
- When evaluating the intrusion sensitivity of a target node i, a node j can send a *query* to node i and obtain the answers. We then use the KNN classifier to assign a value to node i as I_s^i by means of the established model.

In Fig. 2, we give an example to illustrate the assignment of intrusion sensitivity of a node using the KNN classifier. The white point is the incoming feedback waiting for assignment, while based on expert knowledge, we have obtained a set of clusters that are composed of black points (i.e., cluster of *Rate 0.5*). Then, the KNN classifier calculates the Euclidean distance (e.g., ED1, ED2, ED3) between the white-point and the other three clusters respectively. The shorter the distance, the more similar they are. The Euclidean distance between two points can be computed as below:

[3] www.honeybird.hk/

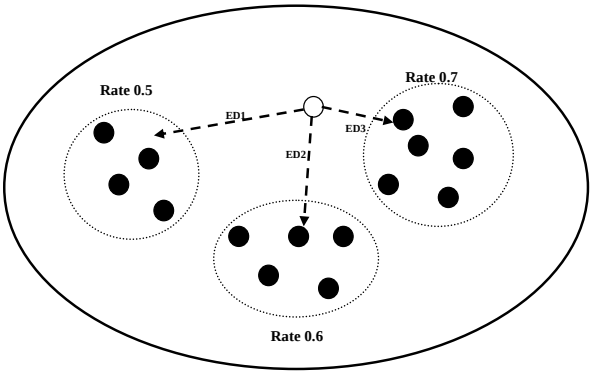

Fig. 2. A case to illustrate the assignment of intrusion sensitivity to a node using the KNN classifier.

$$[Distance\ (P1,\ P2)]^2 = \sum_{0}^{N}(P1_i - P2_i)^2 \tag{4}$$

where $P1_i$ and $P2_i$ are the values of the *ith* attribute of points $P1$ and $P2$ respectively. In real scenarios, each point can be treated as an *answer* to a *query*, and each alarm (and its ranking) in the *answer* can be regarded as an attribute. In this case, we can compute the Euclidean distance between the *current answer* and the *desirable answer*. In Fig. 2, the Euclidean distance between the write point and a cluster (e.g., ED1) can be calculated as below:

$$ED^j_{cluster} = \frac{\sum_1^{N^j_{tn}} ED^i_j}{N^j_{tn}} \quad (i = 1, 2...;\ j = 1, 2...) \tag{5}$$

where $ED^j_{cluster}$ means the Euclidean distance between a target node and a cluster j, ED^i_j means the Euclidean distance between a target node and a node i of cluster j, and N^j_{tn} means the total number of nodes in the cluster j.

Finally, the classifier will find the shortest Euclidean distance and assign the level of *intrusion sensitivity*. For example, if a received answer is classified into one cluster, then the corresponding IDS node will be given the sensitivity level the same as that cluster.

Trust Evaluation of a Node. To evaluate the trustworthiness of a node j, we can use a weighted majority method as follows:

$$T_j = \frac{\sum_{T \geq r} T^{i,j}_{total} D^j_i I^i_s}{\sum_{T \geq r} T^{i,j}_{total} D^j_i} \tag{6}$$

where r is a threshold that node j requests alert ranking to those nodes whose trust values are higher than this threshold. $T^{i,j}_{total} (\in [0, 1])$ is the *overall trust value* of node i according to node j. $D^j_i (\in [0, 1])$ is a measure of *hops* between these two nodes. $I^i_s (\in [0, 1])$ is the intrusion sensitivity of node i.

3.3 Robustness Analysis

The designed CIDN framework and trust management model can achieve good robustness against some common attacks.

Sybil attacks. This attack occurs when a malicious node creates a lot of fake identities. In our trust model, an IDS node should register to a *CA* and obtain a unique proof identity so that our model can defend against this attack.

Betrayal attacks. This attack occurs when a trusted node becomes a malicious one suddenly. Our model employs a forgetting factor in evaluating the trustworthiness so that we can mitigate this attack.

Newcomer (re-entry) attacks. This attack occurs when a malicious node registers as a new user attempting to erase its bad history [20]. But due to our model begins by giving low initial trust values to all newcomers, our model can handle and mitigate this attack.

4 Evaluation

In this section, we present a case study to evaluate the effectiveness of our proposed trust model. The collaborative network, which consists of 30 nodes equipped with Snort, is randomly distributed in a $s \times s$ grid region.

To test the trustworthiness of other nodes in the *partner list*, each node sends out *challenges* and *queries* with an arrival rate μ. Each *challenge* contains 5 alarms for ranking while each *query* contains 10 alarms for ranking. We also have two assumptions. 1) For *challenges*, we assume that an honest node always generates feedback truthfully, while a dishonest node always sends feedback opposite to its truthful judgment. 2) For *queries*, we assume that all nodes will rank the alarms truthfully. Some simulation parameters are shown in Table. 1.

Table 1. Simulation parameters in the experiment.

Parameters	Value	Description
μ	15/day	arrival rate
λ	0.9	forgetting factor
r	0.8	trust threshold
$T_{dir,initial}$	0.5	trust value for new comers
m	10	lower limit of received feedback
s	5	size of grid region
$k1$	5	satisfaction levels
$k2$	10	intrusion sensitivity levels
(W_1, W_2)	(0.7, 0.3)	weight values for T_{total}

D_i^j is anti-proportional to the hops between the nodes in the number of grid steps. The feedback satisfaction is classified as: very satisfied (1.0), satisfied (0.5), neutral (0.3), unsatisfied (0.1), and very unsatisfied (0). The *intrusion sensitivity* (I_s^i) are classified into ten levels such as expert (1.0), excellent (0.9), very high (0.8), high (0.7), good (0.6), neural (0.5), not good (0.4), low (0.3), very low (0.2), and lowest (0.1).

4.1 The Effect of Intrusion Sensitivity: A Case Study

We first evaluate the performance of *intrusion sensitivity* using a metric of *survival rate*, which is defined as the number of nodes which resist the malicious attack divided by the number of all nodes in the network. In this evaluation, we conduct a worm attack to the above network based on [3].

In particular, IDS nodes were running RedHat Linux 7.3 and Apache 1.3.23 web server with OpenSSL encryption enabled. Note that this configuration is vulnerable to the Slapper worm. Later, we launch worm attacks and investigate the survival rate under the situations with and without the *intrusion sensitivity* respectively. We experimented with 1, 3, 5, and 10 protected peers, whereas all other IDS nodes were vulnerable to the worm attack. If this attack hits a protected node, then this node can warn the other nodes for this attack.

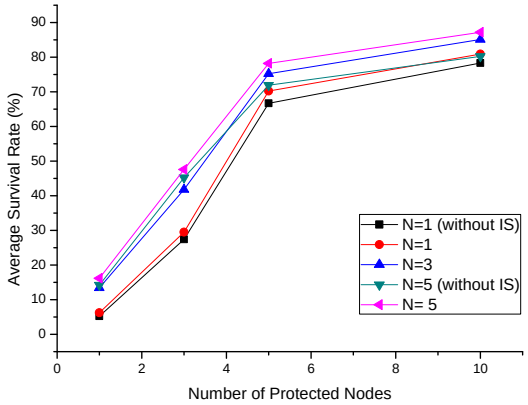

Fig. 3. The results of survival rate.

For each case, we repeated the experiment 10 times during the experiments. In Fig. 3, we illustrate the average survival rates for different configurations, where N means the number of expert nodes that correlates alerts by considering the *intrusion sensitivity*. There are two observations in the experiment:

- This figure shows that the average survival rate increases with the number of protected nodes since more protected nodes can increase the probability of detecting this attack as early as possible.That is, the attack may hit first protected node earlier and this node can warn other nodes more quickly.
- In addition, the average survival rate increases with the number of expert nodes (N) which consider the *intrusion sensitivity*. Taking $N = 5$ for an example, our approach can achieve an average survival rate of nearly 87% while the rate decreases to 80.2% without considering the *intrusion sensitivity*.

In this experiment, we aim to explore the effect of *intrusion sensitivity*. It is found that our approach can achieve a higher survival rate under the attack

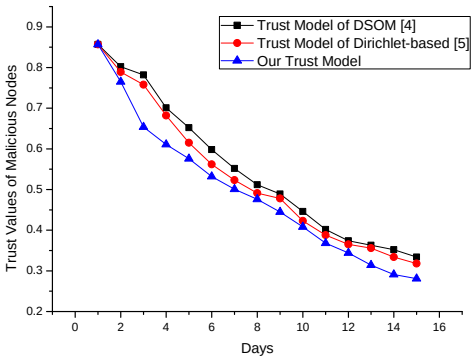

Fig. 4. The trust value of malicious peers.

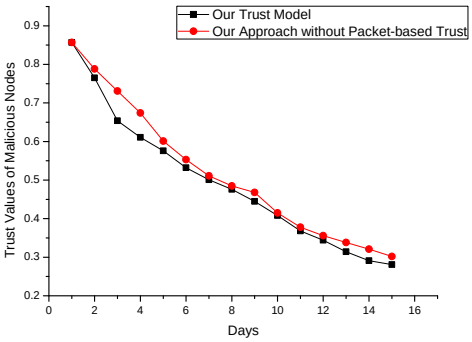

Fig. 5. The effect of packet-based trust on our approach.

scenario by considering the notion of *intrusion sensitivity*. In other words, the experimental results indicate that our approach is promising to help detect malicious attacks by emphasizing the impact of *expert nodes*.

4.2 Defending Against Betrayal Attacks

The goal of this experiment is to study the robustness of our trust model against betrayal attacks, where a malicious node gains a high trust value but suddenly starts to act dishonestly. In addition, we assume that the malicious nodes will launch a port scanning attack to others. We compare our model with two similar models in literature and analyze the effect of packet-based trust on trust evaluation. The comparison results are shown in Fig. 4 and Fig. 5 respectively.

Fig. 4 evaluates the trust values of the betraying nodes after launching the betrayal attacks by means of our model and the trust models of [4] and [5] respectively. The observations are described as below:

- By comparing trust models of [4] and [5], it is found that the Dirichlet-based model [5] can achieve a slight improvement than the model of DSOM [4], since the Dirichlet-based model adopts a dynamic test message rate and can react more swiftly.
- By comparing our model with the other two models, it is visible that our model can make the trust values of malicious nodes drop more quickly. The main reason is that our trust model integrates the *intrusion sensitivity* and depends on two trust types (feedback-based and packet-based trust). Therefore, our model can be more sensitive to react to malicious behavior.

On the other hand, Fig. 5 computes the trust values of malicious nodes under two conditions with and without *packet-based trust* respectively. The observations are described as follows:

- It is noticeable that by considering the packet-based trust, our model can perform better, since the packet-based trust can evaluate the trustworthiness

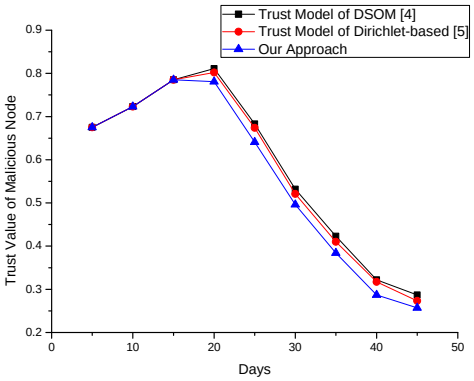

Fig. 6. The trust value of malicious peer.

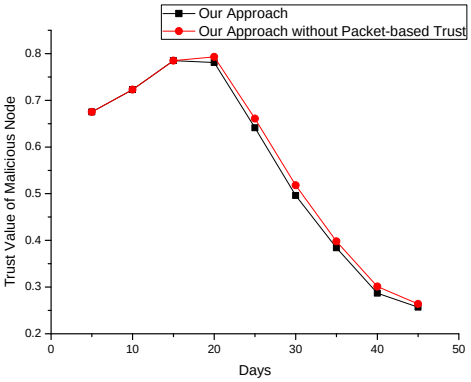

Fig. 7. The effect of packet-based trust on our approach.

of a node in packet level. When a node becomes malicious, it will launch port scanning attack so that these malicious packets can be detected by Snort. In this case, it can improve the detection accuracy of malicious peers and make our model react to malicious behaviors faster by considering the packet-based trust.

– In addition, Fig. 4 shows that our approach can achieve a similar performance without the packet-based trust, as compared to the Dirichlet-based model [5]. However, our approach can still outperform the model of [5] a bit, since our model integrates the notion of *intrusion sensitivity*. This makes our trust model be more sensitive to malicious behaviors.

In this work, once the trust values of malicious nodes drop below the trust threshold of 0.8, these nodes can be ignored and their impact is completely eliminated. The experimental results above demonstrate that our proposed trust model is promising and effective in improving the detection accuracy of malicious nodes as compared to similar models.

4.3 Defending Against Newcomer and Sybil Attacks

Against newcomer attacks. For the newcomer (re-entry) attacks in which a malicious node registers as a new user to erase its bad history, we also conduct an experiment to simulate this situation. It is found that our model is robust against this type of attacks as only a lower initial trust value like 0.5 will be assigned to a newcomer. Due to the initial trust value is lower than 0.8, the newcomer cannot join the trust evaluation of other nodes. The experimental results are shown in Fig. 6 and Fig. 7 respectively.

Fig. 6 shows that the newcomer should first increase its trust values over 0.8 for a period time aiming to join the trust evaluation. However, if this node becomes malicious after its trust value increases to (or over) 0.8, this behavior actually becomes a betrayal attack. Fig. 6 presents that our model is robust against

betrayal attack since the trust values of malicious peers will drop quickly. In addition, Fig. 7 shows that the packet-based trust can improve the performance and robustness of our model in detecting malicious peers.

Against Sybil attacks. Our model is robust to Sybil attacks where a malicious node creates a lot of fake identities, as an IDS node should register to a legitimate *CA* and obtain a unique proof identity. In addition, the trust value of the new joined node is only 0.5 in which the new node cannot make any negative effect on the performance of the network.

5 Challenges and Limitations

We have demonstrated the performance of our model in a simulated environment. In this section, we discuss the challenges and limitations of our current work.

- We acknowledge that the current framework may increase some burden for a node, since it needs to send many messages with other nodes. However, the workload can be predicted as these messages are sent in a period of time. To investigate this issue, we have two directions in our future work: 1) studying the performance of our model with different message arrival rate; and 2) exploring the real burden of communication under our framework.
- We also acknowledge that it is a big challenge to objectively and correctly assign the values of *intrusion sensitivity* based on expert knowledge, as experts may have different views regarding the settings of IDS nodes. Therefore, different levels of intrusion sensitivity may be assigned by different experts. To address this issue, we consider one of the potential solutions is to further specify the criterion for evaluating the *intrusion sensitivity*.
- In this work, we have simulated a CIDN environment during the evaluation. Although it is convenient for us to evaluate the effect of different parameters (e.g., arrival rate) in this simulated environment, it is still a big challenge to test our model in a real environment to investigate its practical performance. We thus consider this as one of our future work.

6 Conclusion and Future Work

A collaborative intrusion detection network (CIDN) is expected to have more power in detecting attacks in which an IDS can collect information and learn experience from other nodes. In this paper, we advocate that each IDS node may have different levels of sensitivity in detecting different types of intrusions. We therefore design a trust management model for CIDNs based on the notion of *intrusion sensitivity* aiming to emphasize the impact of an expert node in identifying malicious nodes. In particular, as a study, we develop an expert knowledge-based KNN classifier that can automatically assign the value of *intrusion sensitivity* to an IDS node. The experimental results under different attack

scenarios show that our approach is more effective and sensitive in detecting malicious peers as compared to other similar trust models.

There are many possible topics in further work. Following work could include discussing the calculation of other trust types such as recommendation trust in the trust management model and verifying the impact of the intrusion sensitivity with even larger experiments. Future work could also include evaluating other classifiers in assigning the levels of intrusion sensitivity and investigating the performance of our model in alert aggregation.

Acknowledgments. We thank all anonymous reviewers for their valuable comments in improving this paper.

References

1. Cai, F., Fugui, T., Yongquan, C., Ming, L., Bing, P.: Grey Theory Based Nodes Risk Assessment in P2P Networks. In: ISPA, pp. 479–483 (2009)
2. Chun, B., Lee, J., Weatherspoon, H., Chun, B.N.: Netbait: a Distributed Worm Detection Service. Technical Report IRB-TR-03-033, Intel Research Berkeley (2003)
3. Duma, C., Karresand, M., Shahmehri, N., Caronni, G.: A Trust-Aware, P2P-Based Overlay for Intrusion Detection. In: DEXA Workshop, pp. 692–697 (2006)
4. Fung, C.J., Baysal, O., Zhang, J., Aib, I., Boutaba, R.: Trust Management for Host-Based Collaborative Intrusion Detection. In: De Turck, F., Kellerer, W. Kormentzas, G. (eds.): DSOM 2008, LNCS 5273, pp. 109–122 (2008)
5. Fung, C.J., Zhang. J., Aib, I., Boutaba, R.: Robust and scalable trust management for collaborative intrusion detection. In: Proceedings of the 11th IFIP/IEEE International Conference on Symposium on Integrated Network Management (IM), pp. 33–40 (2009)
6. Fung, C.J.; Zhu, Q., Boutaba, R., Basar, T.: Bayesian Decision Aggregation in Collaborative Intrusion Detection Networks. In: NOMS, pp. 349–356 (2010)
7. Ghosh, A.K.,Wanken, J., Charron, F.: Detecting Anomalous and Unknown Intrusions Against Programs. In: Proceedings of the 1998 Annual Computer Security Applications Conference (ACSAC), pp. 259–267 (1998)
8. Huebsch, R., Chun, B.N., Hellerstein, J.M., Loo, B.T., Maniatis, P., Roscoe, T., Shenker, S., Stoica, I., Yumerefendi, A.R.: The Architecture of PIER: an Internet-Scale Query Processor. In: Proceedings of the 2005 Conference on Innovative Data Systems Research (CIDR), pp. 28–43 (2005)
9. Janakiraman, R., Zhang, M.: Indra: a peer-to-peer approach to network intrusion detection and prevention. In: WETICE, pp. 226–231 (2003)
10. Li, J., Li, R., Kato, J.: Future Trust Management Framework for Mobile Ad Hoc Networks. IEEE Communications Magazine 46(2), 108–114 (2008)
11. Li, Z., Chen, Y., Beach, A.: Towards Scalable and Robust Distributed Intrusion Alert Fusion with Good Load Balancing. In: Proceedings of the 2006 SIGCOMM Workshop on Large-Scale Attack Defense (LSAD), pp. 115–122 (2006)
12. Li, W., Meng, Y., Kwok, L.-F.: Enhancing Trust Evaluation Using Intrusion Sensitivity in Collaborative Intrusion Detection Networks: Feasibility and Challenges. In: Proceedings of the 9th International Conference on Computational Intelligence and Security (CIS), pp. 518–522, IEEE (2013)

13. Meng, Y., Kwok, L.F.: Adaptive False Alarm Filter Using Machine Learning in Intrusion Detection. In: Proceddings of the 6th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), pp. 573–584 (2011)
14. Meng, Y., Kwok, L.-F., Li, W.: Towards Designing Packet Filter with A Trust-based Approach Using Bayesian Inference in Network Intrusion Detection. In: Proceedings of The 8th International Conference on Security and Privacy in Communication Networks (SECURECOMM), Lecture Notes in ICST 106, Springer, pp. 203-221 (2012)
15. Papadopoulos, C., Lindell, R., Mehringer, J., Hussain, A., Govindan, R.: COSSACK: Coordinated Suppression of Simultaneous Attacks. In: Proceedings of the 2003 DARPA Information Survivability Conference and Exposition (DISCEX), pp. 94–96 (2003)
16. Porras, P.A., Neumann, P.G.: Emerald: Event Monitoring Enabling Responses to Anomalous Live Disturbances. In: Proceedings of the 20th National Information Systems Security Conference, pp. 353–365 (1997)
17. Qin, Z., Jia, Z., Chen, X.: Fuzzy Dynamic Programming based Trusted Routing Decision in Mobile Ad Hoc Networks. In: Proceedings of the 5th IEEE International Symposium on Embedded Computing (SEC), pp. 180–185 (2008)
18. Quercia, D., Hailes, S., Capra, L.: B-Trust: Bayesian Trust Framework for Pervasive Computing. In: Stoen, K. et al. (eds.): iTrust 2006, LNCS 3986, pp. 298–312 (2006)
19. Roesch, M.: Snort: Lightweight Intrusion Detection for Networks. In: Proceedings of the 13th USENIX Conference on System Administration (LISA), pp. 229–238 (1999)
20. Resnick, P., Kuwabara, K., Zeckhauser, R., Friedman, E.: Reputation systems. Communications of the ACM 43(12), 45–48 (2000)
21. Scarfone, K., Mell, P.: Guide to Intrusion Detection and Prevention Systems (IDPS), pp. 800–894. NIST Special Publication (2007)
22. Snapp, S.R., et al.: DIDS (Distributed Intrusion Detection System) - Motivation, Architecture, and An Early Prototype. In: Proceedings of the 14th National Computer Security Conference, pp. 167–176 (1991)
23. Snort. Homepage: `http://www.snort.org/`. (May, 2012)
24. Sun, Y.L., Yu, W., Han, Z., Liu, K.: Information Theoretic Framework of Trust Modelling and Evaluation for Ad Hoc Networks. IEEE Journal of Selected Areas in Communications 24(2), 305–317 (2006)
25. Symantec Corp., Internet Security Threat Report, Vol. 16. (July, 2012) `http://www.symantec.com/business/threatreport/index.jsp`
26. Tuan, T.A.: A Game-Theoretic Analysis of Trust Management in P2P Systems. In: ICCE, pp. 130–134 (2006)
27. Vigna, G., Kemmerer, R.A.: NetSTAT: a Network-based Intrusion Detection Approach. In: ACSAC, pp. 25–34 (1998)
28. Wu, Y.-S., Foo, B., Mei, Y., Bagchi, S.: Collaborative Intrusion Detection System (CIDS): A Framework for Accurate and Efficient IDS. In: Proceedings of the 2003 Annual Computer Security Applications Conference (ACSAC), pp. 234–244 (2003)
29. Yegneswaran, V., Barford, P., Jha, S.: Global Intrusion Detection in the DOMINO Overlay System. In: Proceedings of the 2004 Network and Distributed System Security Symposium (NDSS), pp. 1–17, (2004)