Design of a Simulation Framework to Evaluate Trust Models for Collaborative Intrusion Detection

Carol J Fung Jie Zhang Issam Aib Raouf Boutaba Robin Cohen
David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada
{j22fung, j44zhang, iaib, rboutaba, rcohen}@uwaterloo.ca

***Abstract*—Different trust models have been developed for dealing with possible dishonest behavior and attacks from malicious peer Intrusion Detection Systems (IDSs) in a collaborative Intrusion Detection Network (IDN). For evaluating and comparing these models, this paper introduces a simulation framework that incorporates different components namely expertise model, deception model, attack model, and evaluation metrics. The proposed framework offers flexibility for users to adjust the simulation parameters according to their needs. We then compare three existing trust models in this domain to demonstrate the effectiveness of our framework when used in analyzing their efficiency, robustness and scalability.**

I. Introduction

Intrusion Detection Systems (IDS) identify intrusions by comparing observable behavior against suspicious patterns. They can be network-based or host-based. Network-based intrusion detection systems (NIDS) detect malicious activity by monitoring and analyzing network traffic, while host-based intrusion detection systems (HIDS) detect intrusion by monitoring and analyzing the internal activities as well as network traffics of a computer system. Traditional IDSs 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. IDS collaboration enhances the overall accuracy of intrusion assessment as well as the ability of detecting new intrusion types.

Most existing collaboration networks such as [9], [1], and [8], rely on the assumption that participating IDSs cooperate honestly. However, in such collaborative environments, a malicious (or malfunctioning) IDS can degrade the performance of others by sending out false intrusion assessments. This is especially true when the collaboration is among host-based IDSs because hosts can be easily compromised. To protect an IDN from malicious attacks, it is important to evaluate the trustworthiness of participating IDSs.

Different trust models [2], [5], [6] have been developed for dealing with possible dishonest behaviors and attacks from malicious peer IDSs in a collaborative IDN. Many more trust management models are expected to appear in this domain. A unified testbed would then be beneficial for researchers to analyze and compare these trust models with the purpose of improving their performance.

In this paper, we present a simulation framework for evaluating and comparing trust models in the area of collaborative intrusion detection. In this framework, we simulate and abstract the properties of real-world scenarios in this target area when designing different components, including expertise modeling, deception and attack models, and evaluation metrics. The outcome is a research environment in which users can flexibly adjust parameters for the evaluations of their own trust models and to compare with other existing models using unified evaluation metrics provided along with our framework. We also demonstrate the effective use of our framework for comparing three existing trust models.

II. Simulation Framework Design

Our unified testbed simulates a collaborative Intrusion Detection Network that consists of a number of individual Intrusion Detection Systems. Each IDS communicates with other IDSs in the network through a communication layer. A collaboration layer is built upon Intrusion Detection Systems where an IDS coordinates with other IDSs to gain better intrusion detection performance. A trust management layer upon a collaboration system can further improve the efficiency and robustness of the collaboration system.

In general, a trust management system is composed of three parts: trust evaluation, acquaintance management, and feedback aggregation. We will elaborate these three components in section III. In the rest of this section, we first give an overview of the simulation framework design, and describe each component of the framework subsequently.

A. Overview

Our goal is to build a simulation framework which provides a unified testbed for evaluating and comparing all distributed trust management systems for IDNs. To make our framework general, extendable and easy to use, we adopt a modular architecture. Modules are distinguished by their respective functionalities.

The framework consists of the following components: a core simulation engine, an input interface, an output interface, evaluation metrics, and an IDS model which includes an expertise model, a deception model, an attack model, and a trust management model. Among these components in the IDS model, the trust management model is user-specific and will be implemented by users of this framework. The other components are provided by the framework.

The input interface provides a convenient way for users to set up simulation parameters and run customized experiments.

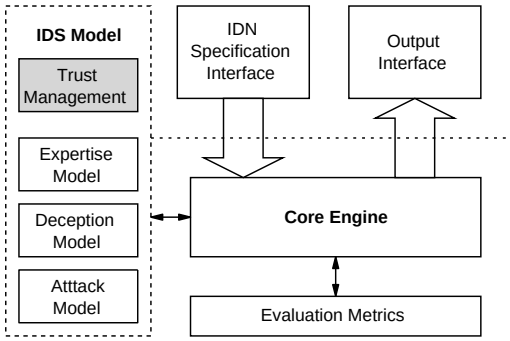

Fig. 1. Framework Design

The parameters include, for example, the size of the IDN, expertise distribution, and adversary strategies. The output interface receives the simulation results data from the core engine and prints out results in graphic mode. The functionalities of other components are described in detail in the following subsections.

B. Core Engine

The core engine is the central part of the simulation framework. Its main functionalities include:

1) Bootstrapping the simulation process and displaying the input interface for users to configure the IDN and experiments;
2) Creating a virtual IDN with a group of IDS instances based on the configurations received from users;
3) Coordinating all the components of the simulation framework to accomplish simulation tasks;
4) Collecting simulation results and sending data to the output interface for plotting.

C. Expertise Model

The purpose of this model is to simulate the environment where IDSs may have different expertise levels in detecting intrusions. Each IDS is assigned an expertise level $l \in (0, 1)$, where a larger l indicates that the IDS is more likely to correctly identify intrusions. In this simulation model, IDSs identify intrusions by ranking the risk levels of alerts $r \in (0, 1)$. The rank of alerts can be, for example, no risk, low risk, medium risk, and high risk. To connect the expertise level of an IDS with the precision of alert ranking, we use a Beta density function to model the possible decisions about alert ranking provided by an IDS with a certain expertise level, as follows:

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

where $f(p|\alpha, \beta)$ is the probability that a peer with expertise level l ranks the risk of an alert with a value of $p \in [0, 1]$. $B(\alpha, \beta)$ is a Beta function, which is a normalization constant for the Beta distribution. We choose the Beta density function because it provides sufficient parameters for us to simulate IDSs with different expertise levels. We define α and β as follows:

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

For a fixed difficulty level $d \in (0, 1)$, the above model has the property of assigning higher probabilities of producing correct rankings to peers with higher levels of expertise. for alerts with higher difficulty levels ($d > l$), A peer with a fixed expertise level l has lower probabilities of producing correct rankings. $l = 1$ and $d = 0$ represent the extreme cases where the peer can always accurately rank intrusions. This is reflected in the Beta distribution by having $\alpha, \beta \to \infty$. Figure 2 shows the probability distribution of the produced risk levels of an alert by peers with different expertise levels, where the true risk level of the alert is fixed to 0.7 and its difficulty level 0.5. The peer with the expertise level of 0.95 has the highest probability of assigning the true risk level to the alert.

D. Deception Models

The simulated IDN environment is populated with adversaries, in order to measure the performance of the trust model being tested by the framework. We introduce four basic deception models: *complementary*, *exaggerate positive*, *exaggerate negative*, and *maximal harm*. In the first three deception models, an adversary may choose to send feedback about the risk level of an intrusion that is respectively opposite to, higher, or lower than the true risk level [10]. We also propose a maximal harm model where a peer always chooses to respond with the intention to cause the worst impact to others.

When an adversary chooses to use a maximal harm deception model, it chooses to report either 0 or 1 for a risk level to achieve maximal deviation from the true risk. In our simulation model, we use a simple threshold decision model for the maximal harm deception model. When the known risk level r is lower than a certain threshold th_m, the adversary always reports the highest risk, otherwise no risk is reported. We plot typical deception curves of deception models in Figure 3. It is worth noting that an adversary node may also adopt a mixture of the basic deception models. It may have an adaptive deception strategy which learns from the environment and accordingly adjusts its deception type and frequency [11]. For example, it may send reports to cause the maximal harm for high risk intrusions and send complementary reports for low risk ones.

E. Attack Models

Adversaries in an IDN may launch attacks to compromise trust models developed in this environment. To evaluate the models' resistance to attacks, the framework integrates a list of common attacks and can be extended to incorporate more attack models.

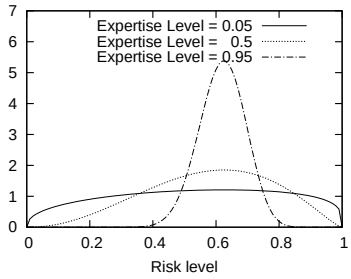

Fig. 2. Expertise Level Model

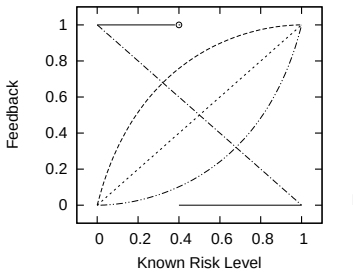

Fig. 3. Basic Deception Models ($th_m = 0.4$)

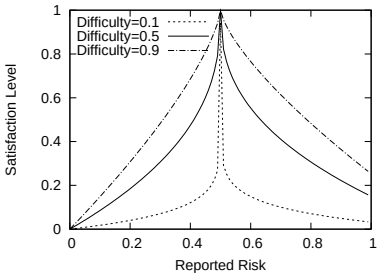

Fig. 4. Satisfaction Mapping Model

1) Newcomer Attack: occurs when a malicious peer whitewashes its bad history by registering as a newcomer, and therefore brings harm to the trust system [7]. Our framework supports newcomer attacks by allowing peers to freely leave and rejoin the network.

2) Betrayal Attack: occurs when a highly trusted peer suddenly changes its behavior to provide untruthful feedback. A trust management system can be degraded dramatically because of this type of attack. In our framework, users can simulate a betrayal attack by specifying a particular time for a certain (expert) peer to become deceptive.

3) Inconsistency Attack: occurs when a malicious peer repeatedly changes its behavior from honest to dishonest, hoping to degrade the efficiency of the trust system without being detected [7]. In our framework, users can choose the frequency and the time interval of honest and dishonest behaviors.

4) Group Attack: occurs when a group of malicious peers launch an attack simultaneously, hoping to achieve a large degression of the trust system. Our framework allows users to specify any number of group attack peers.

F. Evaluation Metrics

This module provides performance metrics, including efficiency, robustness, scalability, and incentives, to evaluate and compare different trust management models. These metrics are computed after simulations and presented to users through the output interface.

1) Efficiency: This metric evaluates the accuracy of a given trust model. It can be represented by the rate of successful detection and the average satisfaction level of aggregated feedback. When a peer encounters an alert it does not know how to rank, it issues a ranking request to its acquaintances. It then aggregates feedbacks to make a final decision. The average satisfaction level of the aggregated feedback will then be used to evaluate the efficiency of the trust model in use. Feedback satisfaction is computed as follows:

$$S(r,a,d) = \begin{cases} 1 - \left(\frac{a-r}{\max(c_1 r, 1-r)}\right)^{c_2 d} & a > r \\ 1 - \left(\frac{c_1(r-a)}{\max(c_1 r, 1-r)}\right)^{c_2 d} & a \leq r \end{cases} \quad (3)$$

where $c_1 > 1$ reflects that the reported risk levels a that are lower than the exact answer r receive stronger penalty than those that are higher. Parameter $c_2 \in (0, 1]$ controls satisfaction sensitivity. Smaller c_2 values yield faster satisfaction decrease when a report deviates from the correct answer. Figure 4 illustrates the mapping function for three intrusion difficulty levels. The chosen parameters are $r = 0.5$, $c_1 = 1.5$ and $c_2 = 0.8$. Notice that the satisfaction level of incorrect answers decreases faster for intrusions with lower difficulty levels (i.e., easy to detect).

2) Robustness: This metric evaluates the robustness of a trust model against attacks that target the trust system. It is indicated by how fast an IDN adopting the trust model can detect these attacks and recover from them. To represent robustness, we observe trust values of attacking peers, the rate of successful detection and the overall satisfaction level of aggregated results within the period when the attacks occur.

A common metric to measure the robustness is the maximum impact of the attacks on the overall efficiency of the network measured by the average satisfaction level of all peers, as discussed in Section II-F1. We can also measure the recovering gap for the average satisfaction level after being attacked. Figure 6 compares the robustness among the example trust models by tracking these two metrics shown as the depth of the pit and the width of the recovering gap respectively.

3) Scalability: This is a critical metric that determines the practical applicability of a trust management model. The most relevant output is the amount of exchanged messages per unit of time per peer denoted as $R(n)$, where n is the total number of peers in the network. Messages here include consultation messages as well as overhead for the purpose of maintaining the trust system. If $R(n)$ increases rapidly with the network size (e.g. $O(n)$), the scalability feature of the model needs to be improved.

4) Incentives: Incentives are important since they influence the long-term effectiveness of a collaborative IDN. A trust management model with a good incentive design can encourage expert peers to contribute more to the network and penalize free-riders [3]. We can measure the amount of help (s_r) that a peer can receive from other peers in the network and the amount of help (s_c) it contributes. In an incentive environment, s_r should be proportional to s_c.

III. Trust Model

We suggest that each trust model tested using our simulation framework includes two major modules: trust evaluation and

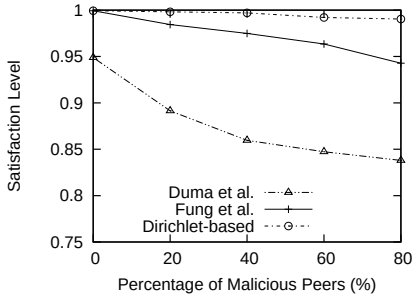

Fig. 5. Efficiency of Trust Models

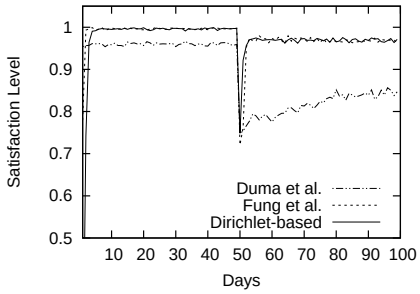

Fig. 6. Robustness of Trust Models

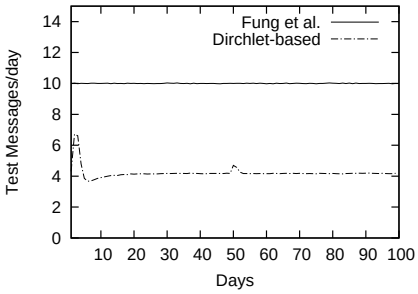

Fig. 7. Scalability of Trust Models

acquaintance management.

A. Trust Evaluation

Trust evaluation is designed to allow a peer IDS to model the trustworthiness of others based on its direct interactions with them in history. For example, the model of Duma et al. [2] assigns a positive interaction with a satisfaction level of 1 and -1 to a negative one. It then uses a linear average of all interactions to calculate the trust value. The model of Fung et al. [5] averages the satisfaction levels of past experience by also incorporating a forgetting factor to emphasize recent experience. The Dirichlet-based trust model [6] adopts a Bayesian model to calculate trust values as well as the confidence of the trust estimation. All trust models are required to have a trust evaluation component.

B. Acquaintance Management

This module is used to manage an acquaintance list for each peer, and decides to which acquaintances and how frequently the peer should send requests. The model proposed in [5] fills each peer's acquaintance list with all other peers in the network. The models in [2] and [6] limit the length of acquaintance lists, keeping only trusted peers and periodically replacing the most untrustworthy peers by new ones. The Dirichlet-based model adopts a dynamic message rate mechanism to allow a peer to send more requests to certain other peers. The other two models have equal message rate for each peer. If a trust model does not have acquaintance management, we assume that each node takes all other nodes as acquaintances.

IV. Demonstration of Our Framework

In this section, we demonstrate the effectiveness of our simulation framework when used to evaluate and compare the efficiency, robustness and scalability of the three example trust models mentioned in Section III, including the model of Duma et al. [2], the model of Fung et al. [5] and the Dirichlet-based model [6]. The simulation creates an environment populated with IDSs with different expertise levels, specifically, low (0.05), medium (0.5) or high (0.95). Simulation using the expertise model is described in Section II-C.

The commonly shared simulation parameters for the following experiments are listed in Table I. The experiment specific parameters will be listed in each experiment description.

TABLE I
Simulation Parameters

Parameter	Value	Description
D	100	Total number of simulation days
c_1	1.5	Cost rate of low estimate to high estimate
c_2	1	Satisfaction sensitivity factor
th_m	0.4	Decision threshold for maximal harm deception

A. Efficiency of Trust Models

This experiment is carried out to demonstrate how to use our framework to evaluate and compare the efficiency of trust models. In this experiment, a peer u has 15 acquaintances, which are evenly divided into three groups with low, medium, and high expertise levels respectively. Among the expert peers, some are malicious and repeatedly adopt the maximal harm deception strategy for two days followed by six days of honest behavior, to degrade the efficiency of the IDN. We also inject intrusions with random risk levels and medium difficulty level (0.5) to peer u in each day. The efficiency of a trust model is measured as the overall satisfaction level of peer u for its aggregated feedback, when the percentage of malicious peers in the network varies from 0% to 80%.

Figure 5 plots the results. We can observe that the Dirichlet-based model outperforms the other two. The dynamic message rate used in the Dirichlet-based model causes the trust values of malicious peers to drop faster and increase slower, and hence minimizes the impact of dishonest behavior. Among the three models, the model of Duma et al. has the lowest efficiency because it does not emphasize recent events and thus responds slowly to sudden changes in peer behavior.

B. Robustness of Trust Models

This experiment demonstrates that our simulation framework can be effectively used for comparing and evaluating the robustness of trust models against various insider attacks. For this purpose, we simulate the betrayal attack. The robustness of each trust model is evaluated by observing how fast the overall satisfaction level of peers can be regained after the attack when the trust model is in use. We set up a scenario where a peer u has seven peers in its acquaintance list, of which six are honest with expertise levels evenly divided between low, medium, and high. The malicious one has high expertise and behaves

honestly in the first 49 days. After that, it launches a betrayal attack by adopting a maximal harm deception strategy.

The results for the satisfaction levels of aggregated feedback with respect to peer u before and after the betrayal attack are shown in Figure 6. We notice that the satisfaction level of u for the aggregated feedback drops down drastically on day 50 and recovers after that in all three models. All three models are similar in the depth of the pit. However, the width of the recovering gap is much shorter for the model of Fung et al. and the Dirichlet-based model. Compared with the model of Fung et al., the Dirichlet-based model has a slight improvement in the recovering speed.

C. Scalability of Trust Models

The result of message rates under betrayal attack is shown for the the trust model of Fung et al. and the Dirichlet-based model in Figure 7, for the purpose of demonstrating the effectiveness of our framework for comparing the scalability of trust models. We notice that in the Dirichlet-based model, the average message rates for highly trusted and highly untrusted peers are the lowest. The average message rate for peers with the medium expertise level is higher. Compared to the model of Fung et al., the average message sending rate is much lower, which demonstrates better scalability of the Dirichlet-based model. Note that the spike from the betraying group on around day 50 is caused by the drastic increment of the message rate. The sudden change of a highly trusted peer's behavior will cause the trust confidence (certainty) level calculated in the Dirichlet-based model to drop down quickly. The rate of sending messages to this peer is then increased accordingly.

V. Related Work

Although deploying a real IDN using existing Intrusion Detection Systems like in [2] can be useful, this type of testbed is expensive to deploy and difficult to unify. It may also lack flexibility for configurations when different aspects of evaluations should be considered. Our simulation framework instead offers a set of parameters that are easy to adjust according to user' needs for different evaluation purposes. Moreover, the performance other than efficiency of trust models including robustness and scalability are incorporated into our framework.

The ART Testbed [4] is proposed to provide unified simulation-based performance benchmarks for evaluating and comparing trust and reputation models in multi-agent systems. This testbed is specifically designed for the e-commerce domain and the only performance metric considered is profit. In contrast, our framework incorporates domain specific knowledge of collaborative intrusion detection (i.e. the extent of the penalty parameter in Equation 3), and introduces performance metrics of robustness and scalability, which are two important concerns in network management.

VI. Conclusions and Future Work

In this paper, we presented a simulation framework for evaluating and comparing distributed trust models in the domain of collaborative intrusion detection. The design of the framework offers flexibility for researchers to adjust parameters that are suitable for the evaluation of their own trust models and to compare with other existing models from different perspectives. We demonstrated the effective use of our framework. Our work thus serves as an initial attempt towards a uniform simulation testbed for trust models in collaborative intrusion networks, with the purpose of allowing researchers in this field to study other models and to improve their own. This work is especially valuable, as IDNs of collaborative IDSs are increasingly used to cope with possible threats.

The current design of our framework targets direct trust models. An indirect trust component that allows peers to ask advice about other peers' trustworthiness may also be incorporated into our framework to reflect reputation of a peer. In this case, other possible attack types (i.e. the bad-mouthing [7]) may also be implemented to allow full testing of trust and reputation models against different attacks. After the extension of our framework is realized, we will implement our framework as an open source to benefit other researchers in this field, and hopefully employing their feedback in order to refine our framework, ultimately resulting in a unified simulation testbed for intrusion detection networks.

References

[1] D. Dash, B. Kveton, J. Agosta, E. Schooler, J. Chandrashekar, A. Bachrach, and A. Newman. When Gossip is Good: Distributed Probabilistic Inference for Detection of Slow Network Intrusions. In *AAAI'06*.

[2] C. Duma, M. Karresand, N. Shahmehri, and G. Caronni. A trust-aware, p2p-based overlay for intrusion detection. In *DEXA'06*.

[3] C. Feldman, M. amd Papadimitriou, J. Chuang, and I. Stoica. Free-riding and whitewashing in peer-to-peer systems. *IEEE JSAC*, 24(5), 2006.

[4] K. K. Fullam, T. B. Klos, G. Muller, J. Sabater, A. Schlosser, Z. Topol, K. S. Barber, J. S. Rosenschein, L. Vercouter, and M. Voss. A specification of the agent reputation and trust (art) testbed: Experimentation and competition for trust in agent societies. In *AAMAS'05*, 2005.

[5] C. Fung, O. Baysal, J. Zhang, I. Aib, and R. Boutaba. Trust management for host-based collaborative intrusion detection. In *DSOM'08*.

[6] C. Fung, J. Zhang, I. Aib, and R. Boutaba. Robust and scalable trust management for collaborative intrusion detection. Technical Report CS-2008-21, University of Waterloo, Canada, 2008.

[7] Y. Sun, Z. Han, and K. Liu. Defense of trust management vulnarabilities in distributed networks. *IEEE Communication Magazine*, Feburary 2008.

[8] Y. Wu, B. Foo, Y. Mei, and S. Bagchi. Collaborative intrusion detection system (CIDS): a framework for accurate and efficient IDS. In *Computer Security Applications Conference, 2003.*

[9] V. Yegneswaran, P. Barford, and S. Jha. Global Intrusion Detection in the DOMINO Overlay System. In *NDSS'04*.

[10] B. Yu and M. Singh. Detecting deception in reputation management. *AAMAS'03*.

[11] J. Zhang, M. Şensoy, and R. Cohen. A detailed comparison of probabilistic approaches for coping with unfair ratings in trust and reputation systems. In *PST'08*.