8-2016

Design and evaluation of advanced collusion attacks on collaborative intrusion detection networks in practice

Weizhi MENG

Xiapu LUO

Wenjuan LI

Yan LI

Published in 2016 IEEE Trustcom/BigDataSE-IPSA, Tianjin, China, pp. 1061-1068.
DOI: 10.1109/TrustCom.2016.0176

Design and Evaluation of Advanced Collusion Attacks on Collaborative Intrusion Detection Networks in Practice

Weizhi Meng*†, Xiapu Luo‡§, Wenjuan Li† and Yan Li¶
*Infocomm Security Department, Institute for Infocomm Research, Singapore
†Department of Computer Science, City University of Hong Kong, Hong Kong SAR
‡Department of Computing, The Hong Kong Polytechnic University, Hong Kong SAR
§PolyU Shenzhen Research Institute, China
¶Singapore Management University, Singapore
Email: {yuxin.meng@my.cityu.edu.hk; csxluo@comp.polyu.edu.hk; wenjuan.li@my.cityu.edu.hk; yan.li.2009@smu.edu.sg}

***Abstract*—To encourage collaboration among single intrusion detection systems (IDSs), collaborative intrusion detection networks (CIDNs) have been developed that enable different IDS nodes to communicate information with each other. This distributed network infrastructure aims to improve the detection performance of a single IDS, but may suffer from various insider attacks like collusion attacks, where several malicious nodes can collaborate to perform adversary actions. To defend against insider threats, challenge-based trust mechanisms have been proposed in the literature and proven to be robust against collusion attacks. However, we identify that such mechanisms depend heavily on an assumption of malicious nodes, which is not likely to be realistic and may lead to a weak threat model in practical scenarios. In this paper, we analyze the robustness of challenge-based CIDNs in real-world applications and present an advanced collusion attack, called *random poisoning attack*, which derives from the existing attacks. In the evaluation, we investigate the attack performance in both simulated and real CIDN environments. Experimental results demonstrate that our attack can enables a malicious node to send untruthful information without decreasing its trust value at large. Our research attempts to stimulate more research in designing more robust CIDN framework in practice.**

***Keywords*-Collaborative Intrusion Detection, Insider Attack, Distributed Network, Collusion Attack, Network Security.**

I. Introduction

As current information infrastructure is vulnerable to various intrusions, intrusion detection systems (IDSs) are widely deployed to protect such resources by identifying malicious actions. Generally, these systems can be classified into two categories based on their detection approaches: misuse-based detection and anomaly-based detection [19]. In particular, a misuse-based IDS detects an intrusion through specifying known signatures of attacks and comparing incoming files to find any match. An anomaly-based IDS detects a potential attack by comparing current behavior with the pre-defined normal behavior (or patterns).

With the rapid evolution of adversary techniques, it is known that a traditional and single IDS, which operates in isolation, is very likely to be compromised by complex or zero-day attacks. To address this issue, collaborative intrusion detection networks (CIDNs) are developed enabling IDS nodes to exchange information with each other [22]. However, in such collaborative environments, a malicious (or malfunctioning) IDS node can make a negative impact on the network performance and degrade alarm aggregation by sending out false intrusion assessments [4]. As a result, it is very important for CIDNs to evaluate the trustworthiness of joined IDS nodes and identify malicious nodes.

Motivations. In the literature, many efforts have been made aiming to protect CIDNs from insider threats (see Section V). Amongst these, challenge-based trust mechanisms (or *challenged-based CIDNs*) are one effective solution [4, 11]. This mechanism evaluates the trustworthiness of a node based on the received answers to the corresponding challenges. A series of studies have proven that challenge-based CIDNs are robust to common insider attacks such as collusion attacks, where a set of malicious nodes cooperate together by providing false alert rankings in order to compromise the network [5, 7]. However, such mechanism assumes that malicious nodes always send feedback opposite to its truthful judgment. In practice, we are aware of that this assumption may be not realistic and can only handle collusion attacks in a weak threat model.

Contributions. As stated above, challenge-based CIDNs are able to defend against collusion attacks under the assumption. But in real-world applications of CIDNs, it is hard to ensure that malicious nodes always send feedback opposite to its truthful judgment. Intuitively, even a malicious node can send truthful feedback pretending to be a benign node. For simplicity, we coin the collusion attacks which are assumed under the assumption to *naive collusion attacks*. In this work, we introduce an *advanced collusion attack* that is practical in real-world networks and evaluate its impact on challenge-based trust mechanism. The contributions of this work can be summarized as follows.

- We first analyze the feasibility of the assumptions used by the existing challenge-based trust mechanisms [4, 7] and point out that this assumption of malicious nodes

Author Note: Weizhi Meng is previously known as Yuxin Meng.

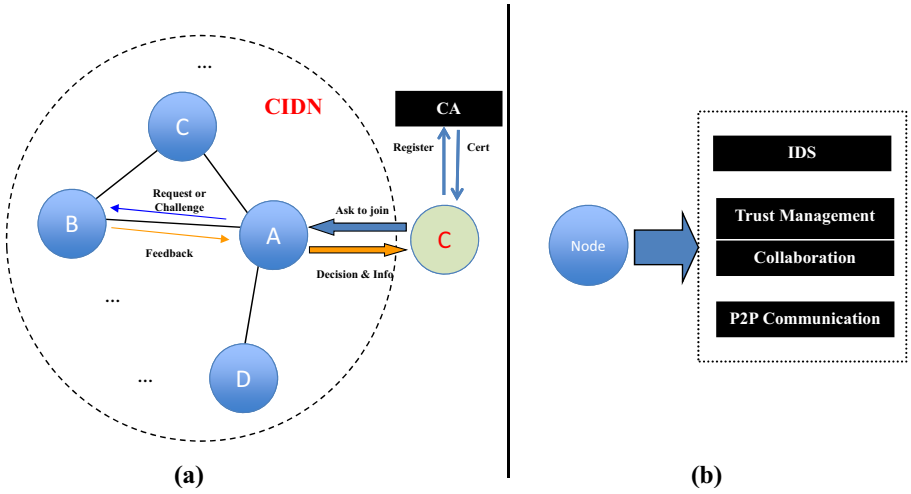

Figure 1. The high-level architecture of challenge-based CIDNs and the major components of a node.

is not realistic in practical implementations. We then develop an *advanced collusion attack*, called *random poisoning attack*, which improves the *naive collusion attack*, in which malicious nodes can choose to send back untruthful feedback in a random manner.

- In the evaluation, we firstly investigate the attack feasibility under a simulated CIDN environment and then explore its performance in a real network environment. Experimental results demonstrate that our attack is effective in practice, where malicious nodes can send false alarm ranking without loosing their trust values at large and keep an impact on alarm aggregation.

To clarify the scope of this paper, we adopt the basic CIDN framework from the literature [7] and limit our discussions to the attack performance. We advocate that challenge-based trust mechanism is an important means to protect CIDNs. Thus, our effort aims to stimulate more research in enhancing existing CIDN architectures and designing more robust CIDN frameworks and trust mechanisms to defend against insider attacks.

The remainder of this paper is organized as follows. In Section II, we introduce the background of challenge-based CIDNs. Section III analyzes the feasibility of assumptions made by the challenge-based trust mechanism and describes our developed *random poisoning attack* in detail. In Section IV, we investigate the attack performance under both simulated and real network environments. Then, we review related work in Section V and conclude the work with future directions in Section VI.

II. Background of Challenge-based Collaborative Intrusion Detection Networks

Generally, challenge-based CIDNs refer to those networks which employ a challenge-based trust mechanism. For better understanding, in this section, we introduce the basic CIDN framework proposed by [4, 5] including major components and the relevant challenge-response mechanism. In Figure 1, we describe the high-level architecture of challenge-based CIDNs and the major components of a node.

In particular, Figure 1 (a) shows how to join the network for a new node and the interactions under the challenge-based trust mechanism.

- *Partner list.* According to [4, 5, 7], an IDS node in the CIDN 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. Such list can be called as *partner list* (or *acquaintance list*). This list is customizable and contains public keys of other nodes and their current trust values.
- *Network join.* To join the CIDN, a node should first register to a trusted certificate authority (*CA*) and get its unique proof of identity (including a public key and a private key). Then, it can ask for joining the network. As shown in Figure 1 (a), if node C wants to join the CIDN, it has to send an application to a CIDN node, say node A. After certificate authentication, node A can send back the decision. If accepted, node C can get an initial list of collaborated nodes from node A.
- *Messages.* In such collaborative network, a node is able to request information from other nodes (i.e., requesting alarm ranking for alarm aggregation). In order to defend against insider attacks, there is another type of messages, called *challenge*, which contains a list of alarms for labeling severity. As a result, there are two types of messages can be sent within a challenge-based CIDN: a *request* for alarm aggregation and a *challenge* for evaluating nodes' trustworthiness.

Figure 1 (b) presents the major components of a CIDN

node including IDS, trust management component, collaboration component, and P2P communication component.

- *Trust management component.* This component is built to evaluate the trustworthiness of other CIDN nodes. According to [4–6], the trustworthiness of a node is mainly computed by evaluating the received feedback. It is worth noting that the testing node should know the correct answers, i.e., severity of the alert described in a *challenge*, so that it can use the received feedback to compute a trust value for the tested node. As stated above, each node can send out *requests* or *challenges* for alert ranking (consultation). To defend against insider attacks, challenges should be sent out randomly and in a way that makes them difficult to be distinguished from requests [4, 5].
- *Collaboration component.* This component is an interface to send out *requests* or *challenges*, and receiving the relevant *feedback*. If an IDS node receives a request or challenge, this component can help send back its feedback as the answers. As depicted in Figure 1 (a), if node A sends a *request/challenge* to node B, node B will send back relevant feedback, respectively.
- *P2P communication.* This component is responsible for establishing connections with other IDS nodes and providing network organization, management and communication between various nodes.

Based on the design, challenge-based CIDNs are capable of identifying malicious nodes under common insider attacks like collusion attacks, where a group of malicious peers cooperate together by providing false alert rankings in order to compromise the network [4, 5]. Under the challenge-based trust mechanism, collusion attacks could be uncovered by means of challenges [5, 11], because they are sent in a random manner and it will be difficult for malicious nodes to distinguish them from actual requests.

III. Our Developed Attack

In this section, we discuss the threat model and relevant assumptions made by challenge-based trust mechanism and introduce an advanced collusion attack, *random poisoning attack*, which can be used to compromise its robustness.

A. Threat Model and Assumptions

As described earlier, challenges will be sent in a random manner under the challenge-based CIDNs and a node is hard to distinguish challenges from requests. These two conditions can be considered as the first assumption for challenge-based CIDNs. The main purposes of such design can be summarized as below:

- The random manner aims to protect *challenges*, since malicious nodes cannot predict when they will receive *challenges*. As a result, to maintain their trust values, nodes have to response to each received message.
- To better protect *challenges*, it is further assumed that challenges are not easy to be distinguished from messages. This condition further forces nodes to response to each received message. Otherwise, malicious nodes can choose not to response to *challenges*, but only give false alarm ranking to *requests*.

In the attack simulation, existing challenge-based CIDNs employ 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. For example, when a malicious node receives a ranking request, it will send feedback "no risk" for an alarm whose real risk level should be "medium", because this feedback can maximally deviate the aggregated result at the sender side. To summarize, the second assumption is that *malicious nodes always send feedback opposite to its truthful judgment for challenges*. This assumption is widely accepted in most research studies and their evaluations such as [4, 5, 7, 10, 11].

In real-world applications, we find that the second assumption is not realistic with the rapid evolution of network threats. A malicious node can choose more complex strategies to affect the network performance. This assumption thus may result in a weak threat model and leave challenge-based CIDNs still vulnerable to advanced insider attacks in real implementations. Due to this, we coin the collusion attacks under the above assumption as *naive collusion attacks*. Next, we develop and introduce an *advanced collusion attack*.

B. Advanced Collusion Attack

In this part, we develop and introduce a type of *advanced collusion attack*, called *random poisoning attack*, by improving the above *naive collusion attacks*. We assume that a malicious node can choose whether to send malicious feedback to received messages, which is more realistic in real computer networks. It is worth noting that challenges have to be sent in random and under a message rate. As the number of messages should be bigger than the number of challenges, the basic idea of our attack is to send malicious feedback under a rate (possibility), attempting to response to *requests* and bypass *challenges*.

Success possibility. Let N_m denote the total number of messages sent each day, N_c denote the number of challenges sent each day, and p denote the possibility of meeting any challenge in the same day. Therefore, we can compute the possibility as below:

$$p = N_c/N_m \tag{1}$$

According to [4, 5], N_c is fixed to keep scalability and reduce network congestion. Thus, a larger N_m can decrease the possibility. In practice, former studies have pointed out that a large amount of IDS alarms would be produced each day [13, 16]. That is, N_m is very likely to be high and p can be rapidly decreased in real network environments. This

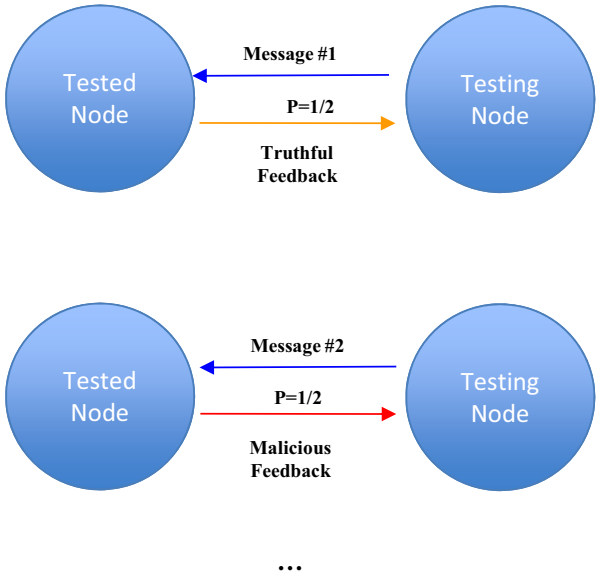

Figure 2. Random poisoning attack on challenge-based CIDNs.

opens a hole for malicious nodes to bypass the examination of challenge-based trust mechanism.

Random poisoning attack. Taking advantage of the vulnerability, we improve *naive collusion attacks* and describe a kind of *advanced collusion attacks*, called *random poisoning attack*, where malicious nodes have a capability of sending malicious feedback (i.e., false alarm ranking) in a random manner. In other words, for a malicious node, the possibility of sending out malicious feedback is $1/2$.

Figure 2 depicts the process of *random poisoning attack*. When receives a message from a testing node, tested node can choose to send back malicious feedback with a possibility $P = 1/2$. In particular scenarios, P can be adjusted (i.e., it is possible to distinguish a challenge from messages). In this work, as we have accepted the condition that a node is hard to distinguish challenges from requests, our attack attempts to use such random manner to compromise the robustness of challenge-based CIDNs.

IV. Evaluation

In this section, we investigate the performance of *random poisoning attack* on challenge-based CIDNs under simulated and real network environments, respectively.

A. Methodology

In the evaluation, we conducted three experiments including two simulations and one real investigation.

- In the first simulation, we aim to explore the performance of *naive collusion attack* under challenge-based CIDNs, where a dishonest IDS node always sends its feedback opposite to its truthful judgement.
- In the second simulation, we try to explore the feasibility of *random poisoning attack*, where a dishonest IDS node can send its untruthful feedback in a random manner (with possibility of $1/2$).
- In the third experiment, we evaluate the attack performance in a real wired CIDN, which is located in an information center. This experiment aims to explore the practical performance of *random poisoning attack* in a real network scenario.

B. CIDN Settings

There are 20 nodes in the simulated CIDN environment, which are randomly distributed in a 5×5 grid region. We use Snort [21] as IDS plugin that can be implemented in a node. Each IDS node can connect to other nodes and establish an initial *partner list* based on the distance. The initial trust values of all nodes in the *partner list* are set to $T_s = 0.5$.

To evaluate the trustworthiness of other partner nodes, each node can send out challenges randomly to other nodes with an average rate of ε. There are two levels of request frequency: ε_l and ε_h. For a highly trusted or highly untrusted node, the request frequency is low, since it should be very confident about the decision of their feedback. But for other nodes, the request frequency should be high because their trust values are close to threshold, thus, we need to monitor them carefully. All the settings are referred to [5, 6, 11]. It is worth emphasizing that we set low request frequency to 10 per day, which is higher and more strict than [5, 6]. The detailed parameters can be summarized in Table I.

Table I
Simulation Parameters in the Experiment.

Parameters	Value	Description
λ	0.9	Forgetting factor
ε_l	10/day	Low request frequency
ε_h	20/day	High request frequency
r	0.8	trust threshold
T_s	0.5	Trust value for new comers
m	10	Lower limit of received feedback
d	0.3	Severity of punishment

Three expertise levels are employed for a node as low (0.1), medium (0.5) and high (0.95). The expertise of an IDS can be using a beta function described as below:

$$f(p'|\alpha,\beta) = \frac{1}{B(\alpha,\beta)} p'^{\alpha-1}(1-p')^{\beta-1}$$
$$B(\alpha,\beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}dt \quad (2)$$

where $p'(\in [0,1])$ is the probability of intrusion examined by the IDS. $f(p'|\alpha,\beta)$ means the probability that a node with expertise level l responses with a value of p' to an intrusion examination of difficulty level $d(\in [0,1])$. A higher value of l means a higher probability of correctly identifying an

intrusion while a higher value of d means that an intrusion is more difficult to detect. In particular, α and β can be defined as [5]:

$$\alpha = 1 + \frac{l(1-d)}{d(1-l)}r$$
$$\beta = 1 + \frac{l(1-d)}{d(1-l)}(1-r) \quad (3)$$

where $r \in \{0, 1\}$ is the expected result of detection. For a fixed difficulty level, the node with higher level of expertise can achieve higher probability of correctly detecting an intrusion. For example, a node with expertise level of 1 can accurately identify an intrusion with guarantee if the difficulty level is 0.

Node Trust Evaluation. To evaluate the trustworthiness of a target node, a testing node can sent a *challenge* to the tested node through a random generation process. The testing node then can compute a score to reflect its satisfaction level. Based on [4], we can evaluate the trustworthiness of a node i according to node j as follows:

$$T_i^j = (w_s \frac{\sum_{k=0}^{n} F_k^{j,i} \lambda^{tk}}{\sum_{k=0}^{n} \lambda^{tk}} - T_s)(1-x)^d + T_s \quad (4)$$

where $F_k^{j,i} \in [0, 1]$ 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* depending on the total number of received feedback, if there is only a few feedback under a certain minimum m, then $w_s = \frac{\sum_{k=0}^{n} \lambda^{tk}}{m}$, otherwise $w_s = 1$. x is the percentage of "don't know" answers during a period (e.g., from $t0$ to tn). d is a positive incentive parameter to control the severity of punishment to "don't know" replies. More details about derivation and feedback satisfaction calculation can be referred to [4, 5].

C. Experiment-1: Naive Collusion Attack

In this simulation, we conduct an experiment to show the robustness of challenge-based CIDNs against naive collusion attack, where a set of dishonest nodes collaborate to send false alarm ranking and always give feedback opposite to its truthful judgment. As thousands of alarms can be generated each day [13, 16], in this simulation, we consider that the number of requests is much bigger than N_c. As a result, we set the $N_m = 100$ in this experiment. The results are shown in Figure 3 and Figure 4.

Figure 3 presents the convergence of trust values for nodes with different expertise levels. It is worth noting that there are three expertise levels: low ($I = 0.1$), medium ($I = 0.5$) and high ($I = 0.95$). In line with the results from [4, 5], nodes with higher expertise can achieve bigger trust values. After around 20 days, the trust of all nodes started converging to stable values.

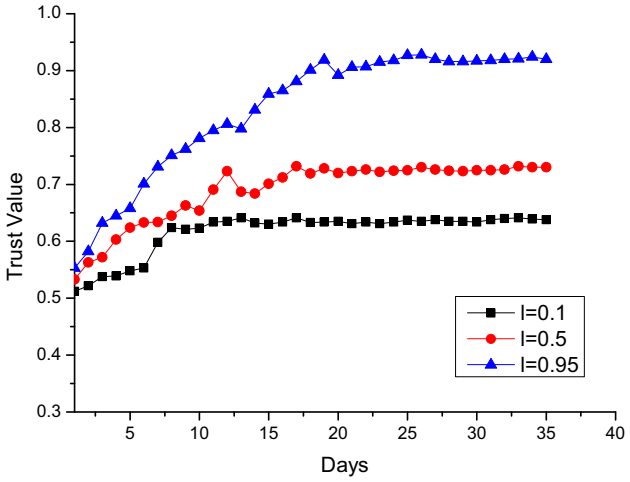

Figure 3. Simulation: convergence of trust values for nodes with different expertise levels.

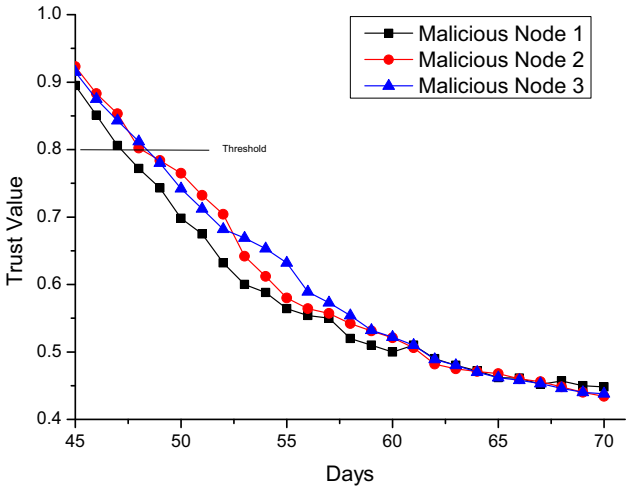

Figure 4. Simulation: trust values of malicious nodes under the naive collusion attack.

To launch naive collusion attack, as a study, we randomly choose three expert nodes ($I = 0.95$) to send untruthful feedback in a constant way from Day 45. We accordingly name these nodes as *malicious node 1*, *malicious node 2* and *malicious node 3*. Figure 4 shows the trust values of these malicious nodes during the attack period. It is noticeable that trust values of these nodes drop quickly below the threshold of 0.8 and cannot make an impact on alarm aggregation. Since malicious nodes always send untruthful feedback to messages including challenges, challenge-based CIDNs can detect such malicious feedback in a short time. The results demonstrate that the challenge-based trust mechanism work well in identifying malicious nodes under the naive collusion condition.

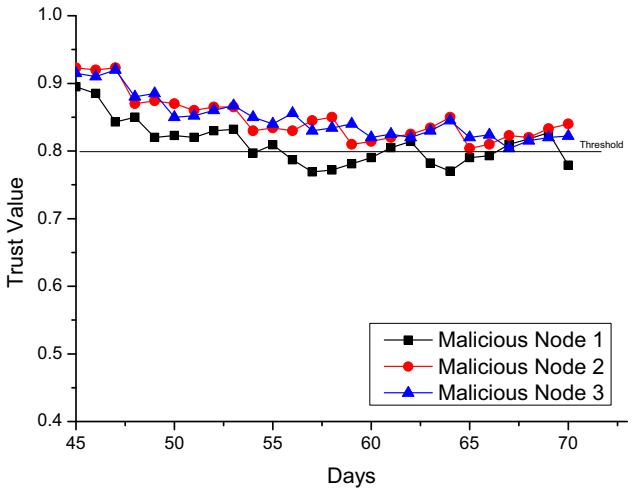

Figure 5. Simulation: trust values of malicious nodes under the advanced collusion attack.

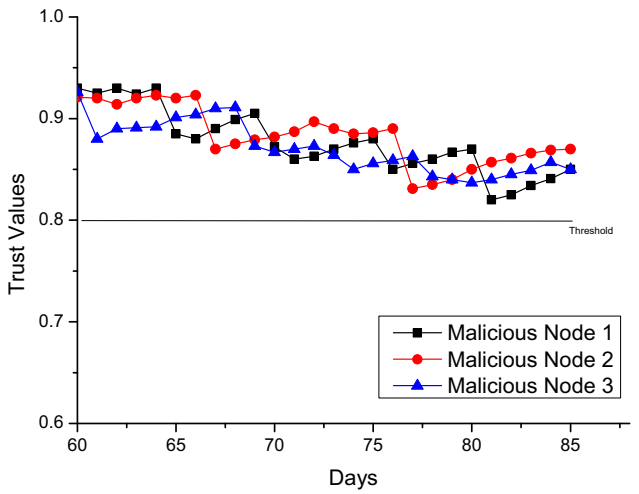

Figure 6. Trust values of malicious nodes under random poisoning attack in a real CIDN.

D. Experiment-2: Advanced Collusion Attack

In this experiment, our main goal is to investigate the feasibility of advanced collusion attack on challenge-based CIDNs. Similar to the above experiment, we also set $N_m = 100$ and use the same expert nodes of *malicious node 1*, *malicious node 2* and *malicious node 3* to launch the attack. The results of trust values are described in Figure 5.

The main observations can be described as follows.

- It is seen that although the trust values of *malicious node 1* and *malicious node 2* were decreasing, their trust values did not fall below the threshold. As a result, they can still make an impact on alarm aggregation.
- For *malicious node 3*, its trust value felt below the threshold during a period from Day 55 to Day 60; however, its trust value increased over the threshold again from Day 61 to Day 63. During the attack period, its trust value always walks around the threshold.

As compared with the results in Figure 4, we find that our advanced attack can prevent a large decease in trust values for malicious nodes through randomly sending back malicious feedback. Even if some malicious answers are detected, these are not enough to quickly reduce the trust values to below the threshold at one go. The results demonstrate the feasibility of our attack.

E. Experiment-3: Random Poisoning Attack in Real Network Environments

In this experiment, we aim to evaluate the practical attack performance in a real and wired CIDN environment. More specifically, this network consists of 25 nodes hosting in an information center, where the incoming network traffic is about 1202 packets/s on average weekly. The basic network settings can be referred to Table I. Differently, as we did not fix the number of messages N_m, this number may be varied each day according to the real daily traffic.

Similar to the above experiments, we first wait the whole network and relevant trust values to be stable. Then, we randomly select three expert nodes as dishonest nodes to launch *random poisoning attack*. The results of trust values are described in Figure 6. The major observations are summarized as follows.

- In the real network, it is found that no malicious nodes can be quickly identified. Actually, none of their trust values are below the threshold of 0.8, so that all of them can still join the process of alarm aggregation.
- Some malicious feedback can still be detected and decrease the trust values of malicious nodes. However, as these nodes only send untruthful responses in a random manner, challenge-based trust mechanism cannot always identify malicious feedback. Hence it cannot quickly reduce the trust values of malicious nodes in a short time. If failed to detect, the trust values of malicious nodes can be recovered gradually.

Overall, the results demonstrate the effectiveness of our attack in a real CIDN environment and reveal that challenge-based trust mechanism should be further improved to handle more complex insider threats in practice.

F. Discussions

In the evaluation, we have demonstrated the feasibility and effectiveness of *random poisoning attack* in compromising the robustness of challenge-based CIDNs in practice. Actually, our attack is just one possible form of advanced collusion attacks in a real network environment. To defend against such advanced attacks, several potential countermeasures can be considered as follows.

- *The number of challenges.* Increasing the number of challenges is an intuitive and possible solution, but this may add more workload on network performance (i.e., causing network congestion). Thus, a balance should be made between challenges and requests.
- *Adding other trust levels.* As challenges should be sent under a fixed message rate, it opens a hole for advanced attackers. To complement this, adding other trust levels is a promising solution. For example, detecting malicious packets in the network [14].

Overall, our study verifies that advanced attackers may behavior more complicatedly in real network environments. Therefore, a realistic threat model should be considered in designing practical and robust CIDN frameworks.

V. Related Work

Distributed IDS architecture. Distributed network infrastructure is very common for IDSs that allows separate nodes to share and exchange information with each other. In 2003, Janakiraman and Zhang [8] proposed a distributed scheme called *Indra*, that allows sharing information between trusted peers in a network to safeguard a peer-to-peer network as a whole against intrusion attempts. Then, Li *et al.* [9] pointed out that most distributed intrusion detection systems (DIDS) relied on centralized fusion, or distributed fusion with unscalable communication mechanisms. Based on this, they utilize the emerging decentralized location and routing infrastructure to construct a new DIDS. Their experimental results showed that the proposed DIDS could greatly outperform the traditional hierarchical system when facing large amounts of diverse intrusion alerts. Several similar studies can be referred to [1, 3, 15, 17, 20, 23].

Collaborative intrusion detection systems and networks. To encourage more collaboration among nodes, collaborative intrusion detection systems (CIDS) was proposed by Wu et al. [22] to enhance the detection capability of a single IDS. More specifically, CIDS employs multiple specialized detectors across different layers such as network layer, kernel layer and application layer. In addition, they implemented a manager for aggregating the alarms from the different detectors to provide a combined alarm for an intrusion. They evaluated the system under a real-world web based application and three classes of attacks (e.g., buffer overflow, flooding and script-based attacks). They showed that CIDS could reduce the impact of missing alarms and false alarms on the performance.

However, one major issue of CIDS is that it assumes that all peers are trusted. This issue often occurs in most distributed IDS architecture, which makes the whole network vulnerable to insider attacks (i.e., betrayal attacks where some nodes suddenly become malicious). To defend CIDNs against insider threats, it requires to establish trust relationship among nodes. For instance, Duma *et al.* [2] proposed a P2P-based overlay for intrusion detection (Overlay IDS) that constructed a trust-aware engine for correlating alerts and an adaptive scheme for managing trust. The former engine is able to drop warnings sent by untrusted or low quality nodes, and the latter scheme could predict nodes' trustworthiness based on their past experiences. This overlay network considers how to integrate trust mechanism, but did not discuss how to handle more complex insider attacks like collusion attacks.

Challenge-based CIDNs. Following the basic idea from Duma *et al.* [2], Fung *et al.* designed a challenge-based trust mechanism for CIDNs, in which the trustworthiness of a node can be computed based on the received answers to the corresponding challenges. At first, they proposed a host-based IDS collaboration framework [4] that enabled each IDS to evaluate the trustworthiness of others based on its own experience by means of a forgetting factor. The forgetting factor gave more emphasis on the recent experience of the peer. Their framework also provided identity verification and created incentives for collaboration amongst them. Then, they improved their framework with a Dirichlet-based model to measure the level of trustworthiness among IDS nodes according to their mutual experience [5]. This model had stronger scalability properties and was robust against common insider threats and the experimental results demonstrated that the new model could improve robustness and efficiency.

As feedback aggregation is a key component in the above trust model, they further applied a Bayesian approach [6] to feedback aggregation in minimizing the combined costs of missed detection and false alarms. They indicated that the Bayesian approach could make an improvement in the true positive detection rate and a reduction in the average cost. They laster summarized their approaches and framework [7]. By adopting their basic CIDN framework, Li et al. [10–12] discusses how to further improve the detection capability through the use of a parameter, called *intrusion sensitivity*, which measures the different expertise of nodes in detecting particular intrusions.

As challenge-based CIDN is effective against many common insider attacks due to its unique design, this work mainly focuses on such type of CIDN framework and analyzes its robustness in real-world applications. We particularly illustrate an advanced collusion attack that can compromise its robustness in practice.

VI. Conclusion

Challenge-based trust mechanisms have been built in the literature to protect CIDNs against insider threats like collusion attacks. However, we identify that such mechanisms rely heavily on the assumption that malicious nodes always send feedback opposite to its truthful judgment. In this paper, we point out that this assumption may be not realistic in practice and result in a weak threat model. We thus design and introduce an advanced collusion attack, called *random*

poisoning attack, where a malicious node has the capability of sending back malicious feedback in a random manner. In the evaluation, our results under both simulated and real network environments demonstrate that our attack enables a malicious node to send untruthful information without decreasing its trust value at large. Our efforts aim to stimulate more research in designing robust trust mechanisms to defend CIDNs against insider attacks in real scenarios.

Future work includes developing other advanced collusion attacks on challenge-based trust mechanism and enhancing existing challenge-based CIDN framework (i.e., adding other trust levels [14]).

ACKNOWLEDGMENT

This work and Dr. Xiapu Luo (as corresponding author) are supported in part by the Shenzhen City Science and Technology R&D Fund (No. JCYJ20150630115257892) and Shenzhen City Special Fund for Strategic Emerging Industries (No. JCYJ20120830153030584).

REFERENCES

[1] B. Chun, J. Lee, H. Weatherspoon, and B.N. Chun, "Netbait: a Distributed Worm Detection Service," Technical Report IRB-TR-03-033, Intel Research Berkeley, 2003.

[2] C. Duma, M. Karresand, N. Shahmehri, and G. Caronni, "A Trust-Aware, P2P-Based Overlay for Intrusion Detection," *In: Proceedings of the 17th International Workshop on Database and Expert Systems Applications (DEXA)*, pp. 692–697, 2006.

[3] R. Huebsch, B.N. Chun, J.M. Hellerstein, B.T. Loo, P. Maniatis, T. Roscoe, S. Shenker, I. Stoica, and A.R. Yumerefendi, "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.

[4] C.J. Fung, O. Baysal, J. Zhang, I. Aib, and R. Boutaba, "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] C.J. Fung, O. Baysal, J. Zhang, I. Aib, and R. Boutaba, "Robust and scalable trust management for collaborative intrusion detection," *In: Proceedings of the 2009 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 33–40, 2009.

[6] C.J. Fung, Q. Zhu, R. Boutaba, and T. Basar, "Bayesian Decision Aggregation in Collaborative Intrusion Detection Networks," *In: Proceedings of the 2010 IEEE Network Operations and Management Symposium (NOMS)*, pp. 349–356, 2010.

[7] C.J. Fung, R. Boutaba, "Design and management of collaborative intrusion detection networks," *In: Proceedings of the 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 955–961, 2013.

[8] R. Janakiraman and M. Zhang, "Indra: a peer-to-peer approach to network intrusion detection and prevention," *In: Proceedings of the 12th IEEE International Workshops on Enabling Technologies*, pp. 226–231, 2003.

[9] Z. Li, Y. Chen, and A. Beach, "Towards Scalable and Robust Distributed Intrusion Alert Fusion with Good Load Balancing," *In: Proceedings of the 2006 SIGCOMM workshop on Largescale attack defense (LISA)*, pp. 115–122, 2006.

[10] W. Li, Y. Meng, and L.F. Kwok, "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, 2013.

[11] W. Li, Y. Meng, and L.F. Kwok, "Design of Intrusion Sensitivity-Based Trust Management Model for Collaborative Intrusion Detection Networks," *In: Proceedings of the 8th IFIP WG 11.11 International Conference on Trust Management (IFIPTM)*, pp. 61-76, 2014.

[12] W. Li and Y. Meng, "Enhancing Collaborative Intrusion Detection Networks Using Intrusion Sensitivity in Detecting Pollution Attacks," Information and Computer Security 24(3), 2016.

[13] J. McHugh, "Testing Intrusion Detection Systems: a Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory," ACM Transactions on Information System Security 3(4), 262-294, 2000.

[14] Y. Meng, L.-F. Kwok, and W. Li, "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)*, pp. 203–221, 2012.

[15] C. Papadopoulos, R. Lindell, J. Mehringer, A. Hussain, and R. Govindan, "COSSACK: Coordinated Suppression of Simultaneous Attacks," *In: Proceedings of the 2003 DARPA Information Survivability Conference and Exposition (DISCEX)*, pp. 94–96, 2003.

[16] T. Pietraszek, "Using Adaptive Alert Classification to Reduce False Positives in Intrusion Detection," In: Jonsson, E., Valdes, A., Almgren, M. (eds.) RAID 2004. LNCS, vol. 3224, pp. 102–124, 2004.

[17] P.A. Porras and P.G. Neumann, "Emerald: Event Monitoring Enabling Responses to Anomalous Live Disturbances," *In: Proceedings of the 20th National Information Systems Security Conference*, pp. 353–365, 1997.

[18] E. Vasilomanolakis, S. Karuppayah, M. Mühlhäuser, and M. Fischer, "Taxonomy and Survey of Collaborative Intrusion Detection," ACM Computing Surveys 47(4), Article No. 55, 2015.

[19] K. Scarfone and P. Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS)," NIST Special Publication 800-94, 2007.

[20] S.R. Snapp 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.

[21] Snort: An an open source network intrusion prevention and detection system (IDS/IPS). Homepage: http://www.snort.org/ (Access on January 2016).

[22] Y.-S. Wu, B. Foo, Y. Mei, and S. Bagchi, "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.

[23] V. Yegneswaran, P. Barford, and S. Jha, "Global Intrusion Detection in the DOMINO Overlay System," *In: Proceedings of the 2004 Network and Distributed System Security Symposium (NDSS)*, pp. 1–17, 2004.

[24] C.V. Zhou, C. Leckie, and S. Karunasekera, "A survey of coordinated attacks and collaborative intrusion detection," Computers & Securirty 29(1), 124–140, 2010.