2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

A Game-Theoretic Approach to Rule Sharing Mechanism in Networked Intrusion Detection Systems: Robustness, Incentives and Security

Quanyan Zhu, Carol Fung, Raouf Boutaba and Tamer Başar

Abstract— **Collaboration among IDSs allows users to benefit from the collective knowledge and information from their collaborators and achieve more accurate intrusion detection. However, most existing collaborative intrusion detection networks rely on the exchange of intrusion data which raises the privacy concern of participants. To overcome this problem, we propose a knowledge-based intrusion detection network, which provides a platform for IDS users to effectively share their customized detection knowledge in an IDS community. An automatic knowledge propagation mechanism is proposed based on a decentralized two-level optimization problem formulation, leading to a Nash equilibrium solution which is shown to be scalable, incentive compatible, fair, efficient and robust.**

I. Introduction

To protect computer users from malicious intrusions, Intrusion Detection Systems (IDSs) are designed to monitor network traffic and computer activities by raising intrusion alerts to network administrators or security officers. Traditional IDSs work independently from each other and rely on downloading new signatures or detection rules from the corresponding security vendor's signature/rule base to remain synchronized with new detection knowledge. However, the increasing number and diversity of intrusions render it not effective to rely on the detection knowledge from a single vendor, since no single vendor can cover all the possible intrusions due to limited labor and available technology. Indeed, vendors usually choose to cover high priority intrusions which may have large influence among their clients or have high risk levels. Collaborative intrusion detection networks (CIDNs) provide a platform for IDSs to take advantage of the collective knowledge from collaborators to improve the overall detection capability and accuracy. However, most existing CIDNs, such as those in [1], [2], [3], [4], and [5], rely on the sharing of intrusion data with others, which raise privacy concerns from the participants. The other way, sharing detection knowledge such as malware signatures and intrusion detection rules, causes less privacy concern.

In reality, expert IDS users, including security analysts, network administrators, and security system programmers, create their own detection rules or customize existing ones to improve detection accuracy specifically for their individual environment [6]. A new detection rule created by one user may be adopted directly by another user if they have similar network/computer configurations. For example, detection rules created for an academic computing environment may be easily adopted by another similar institution; a new intrusion detection rule created to minimize vulnerability of a software can be adopted by others using the same software. An expert user who creates new rules for newly revealed vulnerabilities may share their rules with others who are subject to similar vulnerabilities. Sharing rules among a large group of users can be an effective way to improve the overall security among all users.

In this paper, we leverage the benefit of intrusion detection knowledge sharing and propose a knowledge sharing collaborative intrusion detection network, where intrusion detection knowledge is shared among users who have similar interests in the community. Accordingly, an automatic knowledge dissemination mechanism is proposed to allow users effectively share detection rules with other users without overwhelming their receiving capacities.

The major contributions of this paper are as follows: 1) We develop a rule dissemination protocol based on a decentralized two-level optimization framework, which determines the information propagation rates to each recipient. We set an optimal rule sharing policy for each node and show the existence of a Nash equilibrium in the intrusion detection network. 2) We employ Bayesian learning for each node to estimate the compatibility ratio of others based on the empirical data collected by the node. 3) We design distributed dynamic algorithms to find the Nash equilibrium and perform comprehensive simulations to demonstrate the efficiency, incentive-compatibility, fairness, robustness and scalability of the rule sharing mechanism.

The rest of the paper is organized as follows. In Section II, we describe a knowledge sharing CIDN framework and propose a two-level optimization model to analyze optimal knowledge propagation in the network. In Section III, we discuss Nash equilibrium of the distributed CIDN model and propose practical algorithms to find it. We conduct simulation-based study of the proposed system in Section IV. Finally, we conclude the paper in Section V.

II. Optimal Knowledge Sharing CIDNs

Defense against attackers is a challenging problem since a defender needs to know all possible attacks to ensure network security, whereas an attacker only needs to know a few attack techniques to succeed. It is often impossible for one person or a small group of defenders to know all attack techniques, but it is common to have knowledge about

Q. Zhu and T. Başar are with the ECE Department and CSL, University of Illinois, 1308 West Main St., Urbana, IL, 61801, USA. E-mail:{zhu31, basar1}@illinois.edu; C. Fung and R. Boutaba are with the Cheriton School of Computer Science at University of Waterloo, Ontario, Canada; E-mail: {j22fung, rboutaba}@uwaterloo.ca; The research at the University of Illinois was supported in part by an AFOSR MURI Grant numbered FA9550-10-1-0573, and in part by the Boeing Company through the Information Trust Institute.

some attacks. As a result, the attackers have a significant advantage over the defenders. This motivates defenders to share knowledge with others to overcome their weaknesses. In fact, some open source intrusion detection systems, such as Snort [7] and OSSEC [8], allow users to create and edit detection rules, which provides an opportunity for users to contribute and exchange intrusion detection rules. The purpose of knowledge sharing CIDN is to provide such a platform for users to share their detection rules with others effectively. In [9], an architecture called SMURFEN is proposed, which is built on a Chord [10] peer-to-peer (p2p) communication overlay. In this section, we propose a model for knowledge sharing CIDNs and design an optimal rule sharing mechanism.

A. CIDN Knowledge Propagation Modeling

Knowledge propagation is an essential part of the CIDN system. In this subsection, we describe a system model for a collaborative network comprising a set of n IDSs, denoted by $\mathcal{N}$. In the network, users are allowed to contribute and share rules with others using peer-to-peer communication substrate. A user i propagates new rules to its neighbors, denoted by $\mathcal{N}_i$, with a probability $p_{ij}, j \in \mathcal{N}_i$, to achieve an optimal impact. We let $n_i = |\mathcal{N}_i|$ be the number of neighbors of node i. The communication in the collaboration network is bi-directional, i.e., if node i propagates rules to node j, then node j also propagates rules to node i. We use a matrix $\mathbf{r} = [r_{ij}]_{i,j\in\mathcal{N}}$ to represent the rule propagation rate between nodes in the network and $r_{ij} \in [0, \bar{r}_i], \forall i, j \in \mathcal{N}$, is the rule propagation rate from node i to node j. To make the design robust to DoS attacks, nodes specify maximum sending rate from their neighbors. We denote by $\mathbf{R} = [R_{ij}]_{i,j\in\mathcal{N}}$ the *requested sending rate* from i to j. Note that R_{ij} is controlled by node j and informed to node i. CIDNs require nodes to control their sending rate under the requested rate, i.e., $r_{ij} \leq R_{ij}, \forall i, j \in \mathcal{N}$. To control the communication overhead, an IDS i can set the upper-bound $M_i \in \mathbb{R}_{++}$ on the total out-bound communication rate, i.e, $\sum_{j\in\mathcal{N}_i} r_{ij} \leq M_i$. Denote by $\bar{r}_i$ the *rule contribution rate* from node i. The rule propagation rate from node i to other nodes can not exceed the rule contribution rate $\bar{r}_i$ of node i. Let $p_{ij} \in [0, 1]$ denote the probability that node i sends a rule to node j when such a new rule occurs. Then the probability can be derived from the rule sending and contribution rates, i.e., $p_{ij} = \frac{r_{ij}}{\bar{r}_i}$.

Propagated rules are not all equally useful to their recipients. To capture the metric of relationship on helpfulness, we use a matrix $\mathbf{C} = [C_{ij}]_{i,j\in\mathcal{N}}$ to denote the *compatibility ratio* between two nodes, where $C_{ij} \in [0, 1], \forall i, j \in \mathcal{N}$, representing the probability or likelihood that a rule useful to node i is also useful to node j. Note that the compatibility matrix can be asymmetric, i.e., $C_{ij} \neq C_{ji}$.

Our goal is to devise a system-wide rule propagation protocol so that the rules contributed by all contributors are fairly distributed to other nodes so as to optimize their impact on the system. To achieve this goal, we model our system based on a two-level optimization problem formulation as sketched in Figure 1. At the lower level, an IDS i solves the optimization problem (PPi) where it chooses its propagation rate $\vec{r}_i$ to optimize its public utility function. At the upper level, an IDS i determines the request rate to all neighbors $\vec{R}_i$ from a private optimization problem (Pi). The choice of R_{ji} at the upper level influences the decision-making at the lower public optimization level.

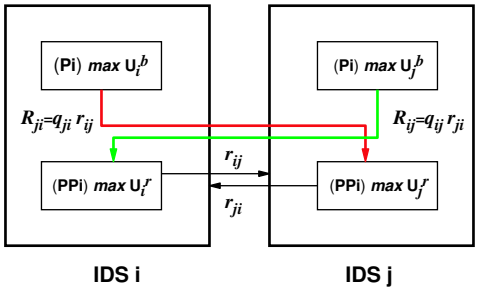

Fig. 1. An illustration of the rule propagation protocol between IDS i and IDS j. Each IDS has a two-level decision process. IDS i optimizes the propagation rate r_{ij} based on an altruistic or public optimization (PPi) and uses a private optimization problem (Pi) to determine the requested sending rate R_{ji} which will be passed on to IDS j for its propagation decisions. It can be seen that the (PPj) decision of IDS j depends on the decision from (Pi) of IDS i. The interdependence of the agents leads to a Nash equilibrium.

B. Lower Level – Public Utility Optimization

In this subsection, we formulate an optimization framework for each node to decide on the propagation rate to all its neighbors to maximize its utility. The utility of each node U_i has two components: a public utility function U_i^r and a private utility function U_i^b. The utility U_i^r measures the aggregated satisfaction level experienced by node i's neighbors weighted by their compatibility ratios. It allows a node to propagate its rules more toward those with whom it is more compatible. On the other hand, U_i^b measures the satisfaction level of a node with respect to the amount of help it receives from its neighbors.

An IDS i can control two sets of variables, $\vec{r}_i = [r_{ij}]_{j\in\mathcal{N}_i}$ and $\vec{R}_i = [R_{ji}]_{j\in\mathcal{N}_i}$. We call $q_{ji} = \frac{R_{ji}}{r_{ij}}$ the greed factor, which reflects the greediness of the request from node j. $q_{ji} > 1$ indicates that node j requests a higher rule propagation rate from node i than the rate it propagates to node i. The introduction of greed factor serves two major purposes: 1) it sets an expectation of return ratio so that a node i can determine its rule propagation rate r_{ij} and R_{ij}/r_{ji} can reach q_{ij} to achieve maximum satisfaction from node j; 2) it serves as an upper bound for communications between nodes i and j, i.e., $r_{ij} \leq q_{ij} r_{ji}$, or equivalently, $r_{ij} \leq R_{ij}$. It circumvents potential denial-of-service attacks from a malicious node who sends an excessive volume of traffic to node j.

The public optimization problem (PPi) seen by each node $i, i \in \mathcal{N}$, is given by

$$(\text{PP}i) \max_{\vec{r}_i \in \mathbb{R}^{n_i}} U_i^r(\vec{r}_i) := \sum_{j\in\mathcal{N}_i} C_{ji} S_{ij}(r_{ij}) \quad (1)$$

$$\sum_{j\in\mathcal{N}_i} r_{ij} \leq M_i, \quad (2)$$

$$r_{ij} \leq R_{ij}, \quad (3)$$

$$0 \leq r_{ij} \leq \bar{r}_i, \quad (4)$$

where $S_{ij} : \mathbb{R} \to \mathbb{R}$ is the satisfaction level of node j in response to the propagation rate r_{ij} of node i. We let S_{ij} take the following form

$$S_{ij}(r_{ij}) := C_{ij} \log\left(1 + \frac{r_{ij}}{R_{ij}}\right). \quad (5)$$

The concavity and monotonicity of the satisfaction level indicate that a recipient becomes increasingly pleased when more rules are received but the marginal satisfaction decreases as the number of received rules increases. The parameter C_{ij} in (5) suggests that a node j is more content when the compatibility or usefulness of rules sent from node i is high.

The objective function $U_i^r : \mathbb{R}^{n_i} \to \mathbb{R}$ in (1) aggregates the satisfaction level S_{ij} of node j by the compatibility factor C_{ji}. The utility U_i^r can be viewed as a public altruistic utility in that a node i seeks to satisfy its collaborators by choosing propagation rates $\vec{r}_i$. The problem (PPi) is constrained by (2) in that the total sending rate of a node i is upper bounded by its communication capacity. The additional constraint (4) ensures that the propagation rate does not exceed its rule contribution rate $\bar{r}_i$. Note that the constraint (3) is imposed by its recipient while constraint (4) is set by node i itself.

Define the sets $\mathcal{F}_i^1 := \{\vec{r}_i \in \mathbb{R}^{n_i} : \sum_{j\in\mathcal{N}_i} r_{ij} \leq M_i, M_i \in \mathbb{R}_{++}\}$ and $\mathcal{F}_i^2 := \cap_{j\in\mathcal{N}_i}\mathcal{F}_{ij}^2$, where $\mathcal{F}_{ij}^2 := \{r_{ij} \in \mathbb{R}_+ : r_{ij} \leq \min(R_{ij}, \bar{r}_i)\}$. The optimization problem is feasible if and only if $\mathcal{F}_i := \mathcal{F}_i^1 \cap \mathcal{F}_i^2$ is not empty. The feasible set is a convex polytope and it can be represented by the convex hull of its finite set of K_i extreme points $\mathcal{K}_i = \{k_1, k_2, \cdots, k_{K_i}\}$, where $K_i = |\mathcal{K}_i|$. Since the utility function (1) is strictly convex in $\vec{r}_i$ and the feasible set is convex, the optimization problem (PPi) is in a form of convex programming and admits a unique solution.

It can be seen that when M_i is sufficiently large and (2) is an inactive constraint, the solution to (PPi) becomes trivial and $r_{ij} = \min(R_{ij}, \bar{r}_i)$ for all $j \in \mathcal{N}_i$. The situation becomes more interesting when (2) is an active constraint. Assuming that q_{ij} and hence R_{ij} have been appropriately set by node j, we form the Lagrangian functional $\mathcal{L}^i : \mathbb{R}^{n_i} \times \mathbb{R} \times \mathbb{R}^{n_i} \to \mathbb{R}$

$$\mathcal{L}^i(\vec{r}_i, \mu_i, \delta_{ij}) := \sum_{j\in\mathcal{N}_i} C_{ji}C_{ij} \log\left(1 + \frac{r_{ij}}{R_{ij}}\right) - \mu_i\left(\sum_{j\in\mathcal{N}_i} r_{ij} - M_i\right) - \sum_{j\in\mathcal{N}_i} \delta_{ij}(r_{ij} - \bar{r}_i), \quad (6)$$

where $\mu_i, \delta_{ij} \in \mathbb{R}_+$ satisfy the complementarity conditions $\mu_i\left(\sum_{j\in\mathcal{N}_i} r_{ij} - M_i\right) = 0$, and $\delta_{ij}(r_{ij} - \bar{r}_{ij}) = 0, \forall j \in \mathcal{N}_i$, where $\bar{r}_{ij} := \min(R_{ij}, \bar{r}_{ij})$. We minimize the Lagrangian (by differentiating it) with respect to $\vec{r}_i \in \mathbb{R}_+^{n_i}$ and obtain the first-order Kuhn-Tucker condition: $\frac{C_{ij}C_{ji}}{r_{ij}+R_{ij}} = \mu_i + \delta_{ij}, \ \forall j \in \mathcal{N}_i$. When (2) is active but (3) and (4) are inactive, we can find an explicit solution supplied with the equality condition

$$\sum_{j\in\mathcal{N}_i} r_{ij} = M_i \quad (7)$$

and consequently, we obtain the optimal solution

$$r_{ij}^\star = r_{ij}^* := \frac{C_{ij}C_{ji}}{\sum_{u\in\mathcal{N}_i} C_{iu}C_{ui}}\left(M_i + \sum_{v\in\mathcal{N}_i} R_{iv}\right) - R_{ij}. \quad (8)$$

When either one of the constraints (3) and (4) is active, the optimal solution is attained at one of the extreme points of the polytope. Since the log function has the fairness property, the optimal solution $r_{ij}^\star$ has non-zero entries when the resource budget is positive, $M_i > 0$. In addition, due to the monotonicity of the objective function, the optimal solution $r_{ij}^\star$ is attained when all resource budgets are allocated, i.e., constraint (2) is active. Hence, the optimal solution $r_{ij}^\star$ to (PPi) is always on the face of the polytope where (7) holds.

Remark 1: We can interpret (8) as follows. The solution r_{ij}^* is composed of two components. The first part is a proportional division of the resource capacity M_i among $|\mathcal{N}_i|$ neighbors by their compatibilities. The second part is a linear correction on the proportional division by balancing the requested sending rate R_{ij}. It is also important to notice that by differentiating r_{ij}^* with respect to R_{ij}, we obtain $\frac{\partial r_{ij}^*}{\partial R_{ij}} = \frac{C_{ij}C_{ji}}{\sum_{u\in\mathcal{N}_i} C_{iu}C_{ui}} - 1 < 0$, suggesting that at the optimal solution, the propagation rate decreases as the recipient sets a higher requested sending rate. If a node wishes to receive higher propagation rate from its neighbors, it has no incentive to overstate its level of request. Rather, a node j has the incentive to understate its request level to increase r_{ij}^*. However, the optimal solution is upper bounded by $\min(\bar{r}_i, R_{ij})$. Hence, by understating its request R_{ij}, the optimal propagation rate is achieved at its boundary point $\min(\bar{r}_i, R_{ij})$.

C. Upper Level – Private Utility Optimization

An IDS i has another degree of freedom to choose its level of requested sending rate R_{ji} of its neighbors. R_{ji} states the maximum rule propagation rate from node j to i that node i can accept. In contrast to the public utility optimization, the optimization at this level is inherently non-altruistic or private. The objective of an IDS i is to choose $\vec{R}_i$ so that its private utility $U_i^b : \mathbb{R}_+^{n_i} \to \mathbb{R}$ is maximized, i.e.,

$$(\text{P}i) \quad \max_{\vec{R}_i\in\mathbb{R}_+^{n_i}} U_i^b(\vec{R}_i), \quad (9)$$

subject to the following constraint from the total receiving capacity $\bar{R}_i$, i.e., $\sum_{j\in\mathcal{N}_i} R_{ji} \leq \bar{R}_i$. Let U_i^b take the form of $U_i^b := \sum_{j\in\mathcal{N}_i} C_{ji} \log(1 + r_{ji}^\star)$, where $r_{ji}^\star$ is the optimal solution attained at (PPi). The log function indicates that an IDS intends to maximize its own level of satisfaction by choosing an appropriate level of request. The request capacity is imposed to prevent excessive incoming traffic as a result of high level of requests. We assume that the capacity is sufficiently large so that the constraint is inactive. Therefore, the decision variable R_{ji} is uncoupled and the problem (Pi) can be equivalently separated into $|\mathcal{N}_i|$ optimization problems with respect to each j, i.e., for every $j \in \mathcal{N}_i$,

$$(\text{P}ij) \quad \max_{R_{ji}\in\mathbb{R}_+} \log(1 + r_{ji}^\star). \quad (10)$$

The following proposition characterizes the optimal choice of R_{ji} or q_{ji} of node i.

Proposition 1: Assume that $\bar{r}_i$ is sufficiently large so that the constraint (4) is inactive. The optimization problem (Pi) admits an optimal solution given by

$$R^*_{ji} = q^*_{ji} r_{ij} = \frac{1}{2} \frac{C_{ij}C_{ji}}{\sum_{u\in\mathcal{N}_j} C_{ju}C_{uj}} \left(M_j + \sum_{v\in\mathcal{N}_j} R_{jv} \right). \quad (11)$$

Proof: The proof of Proposition 1 is in Appendix A. ∎

Combining the solutions to optimization problems (PPi) and (Pi) with the above result, we arrive at

$$r^\star_{ij} = R^*_{ij} = \frac{1}{2} \frac{C_{ij}C_{ji}}{\sum_{u\in\mathcal{N}_i} C_{iu}C_{ui}} \left(M_i + \sum_{v\in\mathcal{N}_i} R_{iv} \right). \quad (12)$$

Equation (12) suggests that an optimal response of node i to node j is to propagate rules at the same rate as the requested rate, which is proportional to the propagation rate sent by node j by the optimal greed factor q^*_{ij} since $R^*_{ij} = q^*_{ij} r_{ji}$.

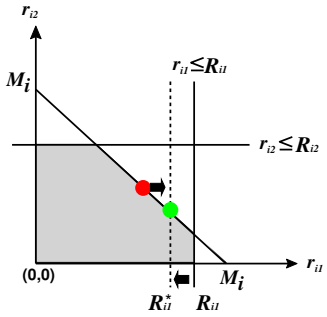

Fig. 2. An illustrative example of a 3-person system involving the set of nodes $\{i, 1, 2\}$. Node i solves (PPi) while nodes 1 and 2 solve (P1i) and (P2i), respectively.

The properties of the solutions to (Pi) and (PPi) are illustrated in Figure 2 for an IDS i and its two neighboring peers. In this illustrative example, we look at the optimal propagation rule for node i to communicate with nodes 1 and 2. Node i solves (PPi) with constraints (1) $r_{i1} + r_{i2} \le M_i$, (2) $r_{i1} \le R_{i1}$, and (3) $r_{i2} \le R_{i2}$. The shaded region is the feasible set of the optimization problem. The optimal allocation can be points on the face of $r_{i1} + r_{i2} = M_i$ of the feasible set. Given the request rates R_{i1} and R_{i2}, suppose the optimal allocation is found at the red point. At the higher level, nodes 1 and 2 need to solve the optimization problems (P1i) and (P2i), respectively. They have incentives to understate their requests. For example, node 1 can request a lower rate until it hits R^*_{i1} and the optimal allocation will increase until it reaches R^*_{i1}. This fact leads to the green point which is the optimal solution to (PPi) found on the vertex of its feasible set given that $r_{i1} \le R^*_{i1}$. Node 2 makes a similar decision and results in R^*_{i2}.

III. Nash Equilibrium and Algorithms

In a collaboration network, each node responds to other nodes by choosing optimal propagation rates and request rates. The two-level optimization problem leads to two game structures of interest. Let $\mathbf{G1} := \langle \mathcal{N}, \{\vec{r}_i\}_{i\in\mathcal{N}}, \{U^r_i\}_{i\in\mathcal{N}} \rangle$ be the game that corresponds to optimization problem (PPi) in which each node chooses its propagation rates given the requested sending rates from its neighbors. Hence, the utilities of the users in Equation (5) reduce to mere functions of r_{ij}. Denote by $\mathbf{G2} := \langle \mathcal{N}, \{\vec{r}_i, \vec{R}_i\}_{i\in\mathcal{N}}, \{U^r_i, U^b_i\}_{i\in\mathcal{N}} \rangle$ the game that corresponds to the two-level optimization problem (PPi) together with (Pi). In $\mathbf{G2}$, each node i chooses its propagation rates as well as its request rates. We next study the existence and uniqueness properties of Nash equilibria (NE) of these two games:

Proposition 2: There exists a NE for $\mathbf{G1}$ and $\mathbf{G2}$.

The proof of Proposition 2 is in Appendix B.

Theorem 1: There exists a NE such that $r_{ij} = R_{ij}$, $\forall i, j \in \mathcal{N}$ in $\mathbf{G2}$. We call such NE a prime NE.

The proof of Theorem 1 is provided in Appendix C. In the following, we state two results on the uniqueness of NE in $\mathbf{G1}$ and $\mathbf{G2}$. Their proofs are in Appendices D and E, respectively.

Proposition 3: Assume that only (2) is an active constraint in optimization problem (Pi) of each node i in $\mathbf{G1}$. Let $\lambda_{ij} = \frac{C_{ij}C_{ji}}{\sum_{u\in\mathcal{N}_i} C_{iu}C_{ui}}$. Then, there exists a unique NE for $\mathbf{G1}$ if $q_{ij}q_{ji} \ne \frac{1}{(1-\lambda_{ij})(1-\lambda_{ji})}$ for each pair of neighbor nodes i, j.

Proposition 4: Assume that $\bar{r}_i$ is sufficiently large and the response of each node follows (12). There exists a unique NE for $\mathbf{G2}$ if $n_i\lambda_{ij} < 2$ for every pair of neighbor nodes i and j.

A. Dynamic Algorithm to Find the Prime NE

Algorithm 1 Distributed Dynamic Algorithm to Find the Prime NE at node i

1: **Initialization** :
2: $\vec{R}^{in} \Leftarrow \{\epsilon, \epsilon, ..., \epsilon\}$ // Small request rates for new neighbors.
3: $\vec{R}^{out} \Leftarrow$ **SendReceive**$(\vec{R}^{in})$ // Exchange requested sending rates with all neighbors.
4: **set** new timer event(t_u, "**SpUpdate**") // Update sending rates and request rates periodically.
5: **Periodic update:**
6: **at timer event** ev of type "**SpUpdate**" **do**
7: // Update the sending rate to all neighbors and then update the requested sending rates from all neighbors.
8: **for** $k = 0$ to B **do**
9: $\vec{r}^{out} \Leftarrow$ **OptimizeSending**$(\mathbf{C}, \vec{R}^{out}, \mathbf{M}, \bar{r})$ // (PPi) optimization.
10: $\vec{r}^{in} \Leftarrow$ **SendReceive**$(\vec{r}^{out})$ // Exchange sending rate with all neighbors.
11: $\vec{R}^{in} \Leftarrow$ **OptimizeRequest**$(\mathbf{C}, \vec{r}^{in}, \bar{R})$ // (Pi) optimization.
12: $\vec{R}^{out} \Leftarrow$ **SendReceive**$(\vec{R}^{in})$ // Exchange requested sending rate with all neighbors.
13: **end for**
14: **set** new timer event(t_u, "**SpUpdate**")
15: **end timer event**

In this subsection, we describe a distributed algorithm (Algorithm 1) for each node to decide on its rule propagation rates. The subscript i is removed for the convenience of presentation. The goal of the algorithm is to lead the system to converge to a prime NE which we defined earlier. In the beginning, nodes set a small requested sending rate for all new neighbors (line 2). An update process is triggered

periodically where function **OptimizeSending** is used for the nodes to find their optimal sending rates $\vec{r}^{out}$ based on the compatibility matrix $\mathbf{C}$ and requested sending rate $\vec{R}^{out}$, which is informed by the acquaintances in process **SendReceive** (line 3). $\mathbf{M}$ and $\bar{r}$ are the sending capacity and rule contribution rate of i, respectively. Function **OptimizeRequest** is used for the nodes to find optimal $\vec{R}^{in}$ ($\mathbf{G2}$) which gives the maximal private utility, given the $\mathbf{C}$, the incoming sending rate $\vec{r}^{in}$, and the receiving capacity $\bar{R}$. The update process is repeated B rounds to yield a converged result.

IV. Evaluation

We simulate a network of n nodes. Each node $i \in \{1, 2, \cdots, n\}$ is labeled with an expertise level $e_i \in [0, 1], \forall j \in \mathcal{N}$, which is the probability that a rule propagated by node i is effective for intrusion detection. Note that the higher the expertise level, the higher the compatibility value. Each node i contributes detection rules to the network following a Poisson distribution with an average arrival rate $\widehat{r}_i$. C_{ij} is learned by j through past experiences using the Bayesian learning method described in [9]. The rule propagation follows the two-level game design described in Section II. In this section, we show some selected results on propagation efficiency, incentive compatibility, fairness, and robustness of the system.

Fig. 3 shows the propagation efficiency for both the mailing list and our system. We define the propagation efficiency to be the percentage of useful rules that nodes receive. We see that when using the our system, the information qualities received by both the low-expertise and the high-expertise nodes are significantly improved compared to the mailing list method. The high-expertise nodes receive higher quality rules than low-expertise nodes, which reflects the incentive-compatibility of the system.

Fig. 4 shows that uniform gossiping provides no incentive to nodes with higher compatibility. On the other hand, the best neighbor propagation scheme provides incentive but no fairness. Nodes of the same compatibility may have very different return benefits. This is because under the best neighbor mechanism, nodes form collaboration groups. Nodes of the same compatibility may join different groups. Since the return benefit largely depends on which group a node belongs to, nodes with the same compatibility values may have significantly different return benefit. On the contrary, our system has a continuous concave utility on the return benefit over compatibility values. It ensures incentive compatibility as well as fairness.

Fig. 5 is to demonstrate the robustness of the system in the face of insider denial-of-service attacks. We can see that the influence of a node is bounded in the system. This is because the system enforces propagation agreements between each pair of nodes. Each node sets a rule propagation limit to all its neighbors using the two-level game (see Section II). Therefore, when a node intends to launch a DoS attack, the amount of rules it is allowed to send to others is bounded by the limits set by its neighbors. Nodes sending excessive traffic to neighbors will be revealed as potential malicious nodes, and thus removed from the neighbor list of others.

V. Conclusion

In this paper, we have studied a rule-sharing collaborative intrusion detection network and used a game-theoretic framework for its protocol design. We have shown that at equilibrium the system has the properties of incentive compatibility, and robustness to denial-of-service attacks. Moreover, the system has also been proved to be fair, efficient and scalable. Through simulations, we have corroborated these important CIDN properties. As future work, we intend to show system robustness to different insider attacks.

Appendix

A. Proof of Proposition 1

From Remark 1, we learn that r^*_{ij} is a monotonic decreasing function with respect to R_{ij} or q_{ij}. Since the utility function in (Pij) is monotonically increasing with r^*_{ji}, increasing R_{ji} will decrease the utility. Hence, an IDS seeks to lower R_{ji} until the optimal utility is achieved to be $U_i^{b\star} = \log(1 + \bar{r}_{ji})$. In other words, an optimal solution R^*_{ji} achieves at $r^*_{ji} = \bar{r}_{ji}$. Assume that $\bar{r}_i$ is sufficiently large, we have $\bar{r}_{ji} = R_{ji}$. Then R^*_{ji} solves

$$R^*_{ji} = \frac{C_{ij}C_{ji}}{\sum_{u\in\mathcal{N}_j} C_{ju}C_{uj}} \left(M_j + \sum_{v\in\mathcal{N}_j} R_{jv} \right) - R^*_{ji}, \quad (13)$$

which yields (11). It is easy to see that any requests $0 < R_{ji} < R^*_{ji}$ will lower the optimal allocation $r^\star_{ij}$ and hence its utility.

□

B. Proof of Proposition 2

In $\mathbf{G1}$, for each $i \in \mathcal{N}$, the feasible set $\mathcal{F}_i$ is a closed, bounded and convex subset of $\mathbb{R}^{n_i}$. The public utility function U_i^r is jointly continuous in its arguments and strictly convex in $\vec{r}_i$. Hence, using Theorem 4.3 in [11], it follows that $\mathbf{G1}$ admits a Nash equilibrium in pure strategies.

In $\mathbf{G2}$, without relaxation, the convex program (PPi) admits a solution $\tilde{r}_{ij}$, which is continuous in $\vec{R}_i$ [12]. The feasible set of (Pi) is compact and convex and the U_i^b is jointly continuous in its arguments and strictly convex in $\vec{R}_i$. Hence, $\mathbf{G2}$ has a Nash equilibrium at the level of private optimization. We can determine $r^\star_{ij}$ which yields an equilibrium at the level of public optimization. Therefore, $\mathbf{G2}$ admits a Nash equilibrium in pure strategies of $\{(\vec{r}_i, \vec{R}_i), i \in \mathcal{N}\}$. □

C. Proof of Theorem 1

We first introduce a few definitions and then prove Proposition 5, which will be used in the proof of Theorem 1.

Definition 1: Let $\vec{R}^*_i, \vec{r}_i, i \in \mathcal{N}$, be a NE. The non-prime degree $\overline{D}$ of an equilibrium is the number of distinct pairs $\{i, j\}, j \in \mathcal{N}_i$, such that $R^*_{ij} \neq r^*_{ij}$. Note that a prime NE has non-prime degree 0.

In this proof, we show that any non-prime NE can be reduced to a prime NE with $\overline{D} = 0$. From Proposition

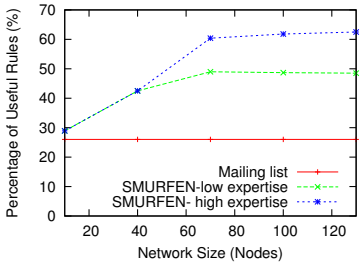

Fig. 3. Propagation Efficiency Comparison

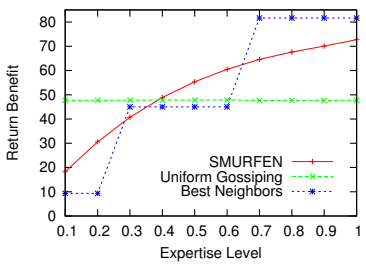

Fig. 4. Incentive on Expertise Levels

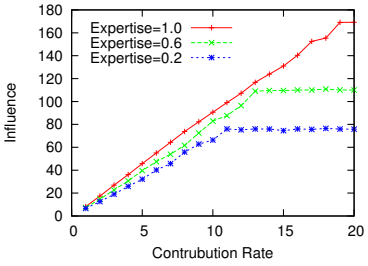

Fig. 5. Influence vs. Sending Rate

2, we know that there exists at least one NE for **G2**. Let $\mathbf{R}^* = [\vec{R}_i^*]_{i\in\mathcal{N}}$ and $\mathbf{r}^* = [\vec{r}_i^*]_{i\in\mathcal{N}}$ be a NE. Suppose it is not a prime NE. Hence, there must exist at least one pair that satisfies $r_{uv}^* < R_{uv}^*$ for some pair $\{u, v\}$. Construct a feasible solution $(\mathbf{R}', \mathbf{r}^*)$ from $(\mathbf{R}^*, \mathbf{r}^*)$ such that $R_{ij}^{'} = R_{ij}^*$, for every $\{i, j\} \in \bigcup_{i\neq j, j\in\mathcal{N}_i, i\in\mathcal{N}}\{i, j\}\backslash\{u, v\}$, and $R_{ij}^{'} = r_{ij}^*$, for $\{i, j\} = \{u, v\}$. From Proposition 5, it follows that $(\mathbf{R}', \mathbf{r}^*)$ also constitutes a NE, whose non-prime degree becomes $\bar{D}_i - 1$. By an iterative process, a non-prime NE $(\mathbf{R}*, \mathbf{r}^*)$ can be reduced to a prime NE. Hence, there exists a prime NE in **G2**. □

Proposition 5: Let $(\mathbf{R}^*, \mathbf{r}^*)$ be a NE with $\bar{D} \neq 0$ and $\{u, v\}$ be a pair of nodes such that $r_{uv}^* < R_{uv}^*$. Let $(\mathbf{R}', \mathbf{r}')$ be a constructed feasible solution such that $\mathbf{r}' = \mathbf{r}^*$, $R_{ij}^{'} = R_{ij}^*$, for every $\{i, j\} \in \bigcup_{i\neq j, j\in\mathcal{N}_i, i\in\mathcal{N}}\{i, j\}\backslash\{u, v\}$, and $R_{ij}^{'} = r_{ij}^*$, for $\{i, j\} = \{u, v\}$. Then $(\mathbf{R}', \mathbf{r}^*)$ is a NE of **G2**.

We need to show that $\mathbf{r}^*$ is an optimal response to $\mathbf{R}^{'}$ and then nodes have no incentive to deviate from $\mathbf{R}^{'}$. For a feasible solution $(\mathbf{R}, \mathbf{r})$, we say that r_{ij} is a boundary allocation if $r_{ij} = \min(\bar{r}_i, R_{ij})$; otherwise, we say that r_{ij} is an internal allocation. At a NE solution, the marginal gains $\frac{\partial U_i^r}{\partial r_{ij}}, j \in \mathcal{N}_i$, are equal for internal allocation points. In addition, the marginal gain of i at boundary allocations is no less than the marginal gains of i at internal allocations.

Since $\mathbf{R}^*$ is a **G2** NE, node v has no incentive to move by changing R_{uv}. If a node v decreases its request to u from value R_{uv}^* to value r_{uv}^*, then the allocation from node u will not increase. This can be easily shown by contradiction as follows.

Suppose the reverse is true, then there must exist an internal allocation r_{um} to m whose marginal gain is higher than the marginal gain at $R_{uv}^{'}$. However, from (2) and (5), we can see that by understating the requests, nodes can increase their marginal gains. Hence, the marginal gain at r_{um}^* is larger than the marginal gain at r_{uv}^*. Therefore, we can conclude that $\mathbf{r}^*$ is not an optimal solution of configuration $\mathbf{R}^*$, which contradicts with the property of NE.

We also observe that node v can not gain from u by either decreasing or increasing its request at $R_{uv}^{'}$. Decreasing the request results in decreasing the allocation from u, since the resource is bounded by the request. On the other hand, increasing the request at $R_{uv}^{'}$ shall not increase the allocation from u, since it will otherwise contradict with the properties of NE $\mathbf{R}^*$ that nodes v can not gain better utility by changing its request at a NE.

Therefore, after the node v decreases R_{uv}^* to $R_{uv}^{'} = r_{uv}^*$, we arrive at $\mathbf{r}^{'} = \mathbf{r}^*$. The constructed solution $\mathbf{R}^{'}$ and $\mathbf{r}^{'}$ is another NE of **G2**. □

D. Proof of Proposition 3

For each pair of collaborative nodes i, j, we have $\mathbf{r}_{ij} = \mathbf{A}_{ij}\mathbf{r}_{ij} + \mathbf{b}_{ij}$, where $\mathbf{r}_{ij} = [r_{ij}, r_{ji}]^T$, $\mathbf{b}_{ij} = [\lambda_{ij}(M_i + \sum_{v\neq j, v\in\mathcal{N}_i} q_{iv}r_{vi}), \quad \lambda_{ji}(M_j + \sum_{v\neq i, v\in\mathcal{N}_j} q_{jv}r_{vj})]^T$, and $\mathbf{A}_{ij} = \begin{bmatrix} 0 & (\lambda_{ij}-1)q_{ij} \\ (\lambda_{ji}-1)q_{ji} & 0 \end{bmatrix}$. Given the existence of Nash equilibrium and the assumptions on q_{ij} and q_{ji}, the uniqueness of the Nash equilibrium is ensured only when $\mathbf{A}_{ij}$ is non-singular. □

E. Proof of Proposition 4

From (12), we can conclude that the optimal response R_{ij}^* to other nodes is given by $R_{ij}^* = \frac{\lambda_{ij}}{2-\lambda_{ij}}(M_i + \sum_{v\neq j, v\in\mathcal{N}_i} R_{iv})$. Since R_{ij}^* is linear in $R_{iu}, u \in \mathcal{N}_i$, we can build the above set of equations into a linear system of equations with the variables $R_{ij}, i, j \in \mathcal{N}$ stacked into one vector. The linear system has a unique solution if the condition of diagonal dominance holds, leading to the condition. □

References

[1] J. Ullrich, "DShield." http://www.dshield.org/indexd.html.

[2] V. Yegneswaran, P. Barford, and S. Jha, "Global intrusion detection in the domino overlay system," in *NDSS'04*.

[3] M. Cai, K. Hwang, Y. Kwok, S. Song, and Y. Chen, "Collaborative internet worm containment," *IEEE Security & Privacy*, vol. 3, no. 3, pp. 25–33, 2005.

[4] C. Fung, O. Baysal, J. Zhang, I. Aib, and R. Boutaba, "Trust management for host-based collaborative intrusion detection," in *19th IFIP/IEEE Intl. Workshop on Distributed Systems*, 2008.

[5] J. Oberheide, E. Cooke, and F. Jahanian, "Cloudav: N-version antivirus in the network cloud," in *Proc. of the 17th USENIX Security Symp.*, 2008.

[6] J. Goodall, W. Lutters, and A. Komlodi, "I know my network: collaboration and expertise in intrusion detection," in *ACM conf. on Computer supported cooperative work*, 2004.

[7] "Snort." http://www.snort.org/ [Last accessed in July 6, 2011].

[8] "OSSEC." http://www.ossec.net/ [Last accessed in July 6, 2011].

[9] C. Fung, Q. Zhu, R. Boutaba, and T. Başar, "SMURFEN: A Knowledge Sharing Intrusion Detection Network," Tech. Rep. CS-2011-06, University of Waterloo, http://www.cs.uwaterloo.ca/research/tr/2011/CS-2011-06pdf.pdf, 2011.

[10] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *ACM SIGCOMM*, pp. 149–160, ACM, 2001.

[11] T. Başar and G. J. Olsder, *Dynamic Noncooperative Game Theory*. SIAM Series in Classics in Applied Mathematics, 1999.

[12] S. Zlobec, *Stable Parametric Programming*. Springer, 1st ed., 2001.