A Reinforcement Model for Collaborative Security and Its Formal Analysis[*]

Janardan Misra
HTS[†]Research
151/1 Doraisanipalya, BG Road
Bangalore 560026, India
janardan.misra@honeywell.com

Indranil Saha[‡]
Computer Science Department
University of California
Los Angeles, CA 90095, USA
indranil@cs.ucla.edu

ABSTRACT

This paper presents a principled approach to one of the many little-studied aspects of computer security which relate to human behavior. Advantages of involving users who usually have strong analytic ability to detect violations and threats but not primarily responsible for security have been well emphasized in the literature. In this work we propose a reinforcement framework for enabling collaborative monitoring of policy violations by the users. We define a payoff model to formalize the reinforcement framework. The model stipulates appropriate payoffs as reward, punishment, and community price according to reporting of genuine or false violations, non-reporting of the detected violations, and proactive reporting of vulnerabilities and threats by the users. We define probabilistic robustness property of the resulting system and constraints for economic feasibility of the payoffs. For estimating the parameters in the payoff model, system and user behaviors are modeled in terms of probabilistic finite state machines (PFSM) and likelihood of the success of the model is specified using Probabilistic Computation Tree Logic (PCTL). PRISM model checker based automated quantitative analysis elicits the process of the estimation of various parameters in the model using PFSMs and PCTL formulas.

Categories and Subject Descriptors

K.6.m [**Management of Computing and Information Systems**]: Miscellaneous—*security*; H.1.2 [**Models and Principles**]: User/ Machine Systems—*human factors*; J.4 [**Social and Behavioral Sciences**]: Psychology; K.4.3 [**Computers and Society**]: Organizational Impact—*computer supported collaborative work*

General Terms

Security, Human Factors

Keywords

Collaborative Security, Collaborative Monitoring and Reporting, Reinforcement Model, Reward-Punishment Framework, Probabilistic Modeling and Analysis

1. INTRODUCTION

With the increasing size of today's organizations having dynamically changing asset base (physical and logical), designing appropriate security policies and their enforcement to maintain confidentiality and integrity of these assets are becoming increasingly difficult. One of the noticeable limitations of the existing security frameworks is that user base of assets is differentiated from the security administrators who design and enforce the security policies. Therefore, it appears a natural proposition that if securing confidentiality and integrity of certain types of assets is considered as a collective responsibility of the users and security administrators, the security enforcement would enhance positively. For example, a malicious user making destabilizing changes in a code base could be better monitored and reported for doing so by the associated team members, who have probably better knowledge of it or can better detect it than the centrally administered monitoring mechanisms.

To make users responsible for the security of the assets (in particular critical assets), a plausible approach may be to involve them in different aspects of security including threat perception and monitoring the violations of policies. Now-a-days, all these operations are mainly taken care by a limited group of administrators. They define security policies, devise means to enforce them, and monitor continuously to detect possible violations. However, a large enterprise-wide organization typically has tens of thousands of employees and many more roles/tasks/permissions, and even larger number of assets and contexts present at any point of time. Thus, understanding the multitude of security requirements and their enforcement for a large organization is not only difficult but also error-prone. It would be a better solution, if different groups formed based upon business focus, roles, emerging contexts, and tasks also participate in defining security policies and are entrusted with collective monitoring of the policy violations. In early 90s, Greenwald [14] advocated similar philosophy in the context of distributed resource management and access control and proposed a Distributed Compartment Model, which allows users to manage resources across different administrative domains with increased independence from central system administrators. Also Vimercati and Samarati [10] proposed a model with local user autonomy in access control for federated databases.

In this paper we consider the problem of collaborative enforcement and monitoring of security policies. To guide individuals and groups for this, there needs to be a well-defined framework.

[*] Patent applications pending in USPTO: 12/057855 and 12/171225.

[†] Honeywell Technology Solutions

[‡] The work has been carried out when this author was a member of HTS Research

NSPW'09, September 8–11, 2009, Oxford, United Kingdom

This framework should be easy to follow for devising measures to ensure overall implementation of such collaborative monitoring efforts. Also as an organization's policies change over time, the framework should be such that it can effectively adapt with the changes. Unfortunately existing models of security do not consider such collaborative aspects and thus there is a need to devise one such.

In this work, we present a formal framework for devising policies to enable collaborative monitoring against policy violations. Importantly, presented framework does not imply that the employees take up the additional roles of security completely, however in certain scenarios, where they could have more effective role in enforcing the policies and are directly impacted by the violations, it is indeed desirable that they take proactive participation. A few such representative scenarios are discussed next to motivate the need for having a model for collaborative monitoring by the users:

Sensitive Data Manipulation: Suppose a (disgruntled) employee or a group of employees has privileged shared access to sensitive data or device, e.g. strategic documents, design documents, source code, or sensitive infrastructure control and attempts to manipulate the data or device by introducing disruptive changes. Chances are higher that other users (e.g. supervisors) also having access rights would be able to detect such disruptive manipulation since they have the required semantic understanding to determine the potential impact owing to such changes with the contents.

Photo ID cards – Changing identities: Now-a-days most of the corporate organizations provide their employees with photo printed smart cards, which enable access to different facilities around. However, in a large organization, it might not be possible for the limited number of security staffs to monitor if everyone present in the organization is indeed using his/her own access cards. And if an intruder is somehow able to get such a smart card, it could enable his access to the facilities and could potentially cause serious threats. However, it is more likely that when such intruder tries to access these facilities, other users familiar with the facility might notice and report the presence of such unfamiliar intruder than his getting detected by an existing monitoring infrastructure.

Informal transfer of sensitive IP: Suppose Alice and Bob, being part of an R&D department work on some sensitive projects. During their visit to a scientific conference, Foe, a friend of Bob, working for a competitor organization meets Bob and they discuss their ongoing research, where Alice happens to join them. Alice may notice that their discussions might amount to disclosure of the crucial IP, not yet legally protected. Alice on noticing this can bring it to the notice to authorities and help the organization in protecting the IP as soon as possible.

A discussion on the deliberate coding violations by programmers and IP theft also appears in [15] highlighting the potential loss which such violations may cause to the organizations especially in the context of safety critical applications.

In these scenarios a social framework always provides wider scope and depth for monitoring the violations than any existing monitoring infrastructure. This is especially true for the context dependent logical resources, i.e., some data, significance of which is realized only when considered with respect to specific contexts, e.g. design documents and source code having product specific interpretation and significance. Such resources are most vulnerable to 'semantic manipulations' and securing them using automated monitoring is either not feasible or would be very costly.

A special class of such threats is known as *insider threat* [8, 38, 18, 4, 3], which is a pressing problem for most of the organizations today. Recent studies on insider threats in finance and banking sector indicate that often other users (85%) had some amount of prior knowledge of the possible threat [33]. Indeed, in 61% of the cases, insiders responsible for the threats were actually detected by the people who were not responsible for the security. In another study, it was estimated that more than 25% of the frauds were actually reported by non-security users involving customers and other coworkers [6].

A collaborative monitoring could then be considered as a kind of social networking based monitoring mechanism whereby each member having access on shared resources is expected to monitor for the compliance and specifically report the instances of violations of the associated security policies or potential threats. The fundamental question, which arises in such a scenario, is as to how can such a collaborative monitoring framework be made effective since there may not exist any prior (latent) positive *network effect* for users to monitor against the security violations. The network effect is often considered a fundamental prerequisite for the success of any collaborative networking phenomenon. Network effect [25, 21, 29] is generally expressed as a utility function which determines some useful value compared to the price paid for an actively engaged user in a network in terms of the size of the network. For a positive network effect such a utility function must be non decreasing w.r.t. the number of users in the network for a sufficient range.

Drawing inspiration from the organic unity represented in the biological systems against attacks and socio-psychological studies on security and human motivation, in this work, we propose a reward-punishment based reinforcement framework for enabling collaborative monitoring of policy violations by extrinsically inducing positive network effect in the system. To this aim, the framework stipulates appropriate payoffs as rewards, punishments, and community price according to the reporting behavior of the users on genuine or false violations, non-reporting of the detected violations, and proactive reporting of potential vulnerabilities and threats. We use a payoff matrix based mechanism to formalize the framework.

The rest of the paper is organized as follows: In Section 2 we set the background context for the model including assumptions and socio-psychological studies on extrinsic motivation. Section 3 formally elaborates the payoff model followed by discussion on implementation issues in Section 4. In Section 5 we present a probabilistic model for analyzing the collaborative monitoring behavior under proposed framework and discuss experimental work using PRISM model checker in 6. Section 7 presents a discussion on the potential limitations of the presented framework and Section 8 concludes the article.

2. THE FRAMEWORK

Before we discuss the model further, let us specify the underlying assumptions:

2.1 Assumptions

Observability: Proposed model assumes that all genuine occurrences of policy violations have an 'observable impact' on the system, which could be determined by the security administrators. Thus we only consider such violations, which affect the state of the system and do not consider other kinds of 'silent' violations not affecting the system as far as the observable state of the system is concerned. This in turn, implies that security administrators will always be able to identify and validate the occurrence of a violation even if it remains undetected by the users. This assumption thus avoids the cases of false positives as further discussed in the Section 7.

Detection: A violation is considered to be detected only when it is reported to be done so (either by users or some monitoring device). Therefore if a violation occurs but is not reported by any of the witnesses (or captured by the monitoring device), it would be considered undetected.

Policy Synthesis: Model assumes that security policies are defined a priory. Nonetheless, it is possible that as a by product of the monitoring process, existing policies are refined or new policies may potentially be integrated into the framework as determined by the existing policy synthesis machinery. For example, certain sequence of events (each event is an operation on an object by some subject) might enable other access restriction violations, and therefore reporting the final access violation in terms of the scenarios consisting of these sequence of events might give rise to new set of access restrictions.

Policy Completeness and Consistency: Policies are also assumed to be contradiction free, mutually consistent, and complete. Lack of contradiction in a policy definition and mutual consistency among policies are required to avoid the cases of ambiguity in their interpretation by the users and for determining whether a scenario should actually be considered as a violation or not. On the other hand, the policy completeness requirement ensures that decisions could be made in all related scenarios. We discuss more on this assumption in Section 7 on 'Challenges in Collaborative Security'.

Policy Awareness: Model assumes that users have necessary knowledge of legitimate accesses/policies and capability to detect and report genuine violations. In the beginning, however, users may not have complete knowledge of the policies and with time, as violations would be reported, these would reinforce the awareness of the users. We will discuss more on this assumption in Section 4 on 'Implementation Issues'.

2.2 Socio-psychological Dimension

The justification to externally induced network effect comes from the numerous studies in social psychology on the role of extrinsic motivation in affecting individual and group attitudes and behaviors [37, 30, 9, 11]. These studies provide insights into what are the usual behavioral effects of various kinds of rewards and punishments. Some of the conclusions from these studies are quite important while we discuss in the next section the mechanism for collaborative monitoring. These are discussed next:

1. Extrinsic rewards can be important motivator to start new (community) behaviors in the individuals.

2. Group punishment mechanisms usually play an important role in the continuation of the intuitively justified community behaviors. Individuals in groups tend to exert pressure (though not always explicitly) on other individuals to avoid themselves from paying community punishments owing to the violations caused by others.

3. Apart from rewards, punishments are also used as negative reinforcement tools for the individuals, who try to avoid such punishments by following the expected behaviors. Nonetheless, unless expected behaviors have been internalized by the individuals, the withdrawal of such negative reinforcements may put individuals at the risk of reverting to the old situation.

4. Sociological studies on the concept of *locus of control* [34, 35] reveal that individuals show increased motivation towards activities when they perceive better control over their environment. In essence collaborative security lets users have a say in designing policies and monitoring their violations, which would give them a sense of better control over the assets and policies they are using and in turn over the security environment against the current scenario where they have little or no say on these aspects.

Kabay [19, Chap. 35] also discusses the importance of applying socio psychological understanding of individual and group behavior while designing security policies. For example, he emphasizes on the need for having policies and environment of rewarding employees for reporting security violations.

Based upon this understanding, we will next formally define the payoff matrix model as an enabling mechanism for the collaborative monitoring.

3. THE PAYOFF MATRIX MODEL

Let us consider a system consisting of subjects (processes/users) accessing shared resources according to specific (security) policies. The policies may specify that an object has some access restrictions (e.g. copy operation on a specific File not allowed) or may direct the behavior of the subjects (e.g. a user must not share her password).

Formally, let the set of subjects be

$$S = \{s_1, s_2, \ldots, s_n\}$$

and let there be finitely many [1] ways to violate a security policy resulting into a set of violations as

$$Vio = \{vio_1, vio_2, \ldots, vio_m\}$$

We do not assume that policies remain fixed. If a policy changes, that would get reflected in the set of associated violations.

Let us associate with each subject, two types of time varying payoff matrices for each relevant policy violation. These are depicted in Fig. 1 and Fig. 2.

Notations: All the entries in the tables are functions of time implying that their actual value, at any time, might be dependent upon the previous events or past behaviors of the players. t is the time variable with granularity of reporting occurrences. Further,
$R_{ij}(t)$: Reward for player s_i on reporting true primary violation vio_j.
$CP_j(t)$: (absolute value) Community price associated with true primary violation vio_j.
$P'_{ij}(t)$: (absolute value) The payoff for player s_i for not reporting true primary violation vio_j.
$\Theta_{ij}(t)$: Reward for player s_i on reporting potential violation (or threat) on vio_j.
$P_{ij}(t)$: (absolute value) The payoff for player s_i for false reporting on violation vio_j.
$r_{ij}(t)$: Reward for player s_i on reporting true secondary violation on vio_j.
$cp_j(t)$: (absolute value) Community price associated with true secondary violation on vio_j.
$p'_{ij}(t)$: (absolute value) The payoff for player s_i for not reporting true secondary violation on vio_j.
$\partial_{ij}(t)$: Reward for player s_i on reporting potential secondary violation on vio_j.
$p_{ij}(t)$: (absolute value) The payoff for player s_i for false reporting of a secondary violation on vio_j.
#: Undefined value.

The first payoff matrix in Fig. 1 defines the payoffs associated with the i^{th} player s_i for her reporting behavior on j^{th} policy violation vio_j.

[1] In cases where Vio is an uncountable set, suitable equivalence relation needs to be defined which could partition Vio into finitely many classes such that all the violations in each class could be considered equivalent for defining payoffs.

Primary Payoffs	True Violations	False Violations
Reported	$R_{ij}(t)$	- $P_{ij}(t)$
Non Reported + Undetected	-$CP_j(t)$	#
Detected + Not Reported	-$P`_{ij}(t)$	#
Threat Reporting	$\Theta_{ij}(t)$	#

Figure 1: The Payoff table for the Reporting behavior on Primary Violations

Secondary Payoffs	True Violations	False Violations
Reported	$r_{ij}(t)$	- $p_{ij}(t)$
Non Reported + Undetected	0	#
Detected + Not Reported	-$p`_{ij}(t)$	#
Threat Reporting	$\partial_{ij}(t)$	#

Figure 2: The Payoff table for the Reporting behavior on Secondary Violations

We treat non-reporting of a policy violation itself to be a violation, which may invite punishment. We argue that in the absence of such treatment it might not be possible to give rise to a dynamically evolving and increasingly secure system. Therefore second payoff matrix in Fig. 2, defines the payoffs associated with the i^{th} player s_i for the j^{th} policy violation vio_j on the reporting behavior of s_i for non reporting of vio_j by some other player.

Formally, let us consider the primary and secondary the payoff matrices for the subjects against each policy violation:

$$\langle(\bar{P}T_1, \bar{S}T_1)\ldots(\bar{P}T_n, \bar{S}T_n)\rangle$$

where each player s_i is associated with primary payoff tables

$$\bar{P}T_i = [T^P_{i1}, T^P_{i2}, \ldots T^P_{im}]$$

and secondary payoff tables

$$\bar{S}T_i = [T^S_{i1}, T^S_{i2}, \ldots T^S_{im}]$$

such that T^P_{ij}, T^S_{ij} denote the payoff tables corresponding to policy violation vio_j.

In Fig. 1 on Primary Payoffs, first column - *True Primary Violation* represents the case when a *genuine violation* of a policy has occurred – impact of which is assumed to be observable later on. The second column - *False Primary Violation* represents false violations where player s_i may act on the basis of a *fabricated violation* – a violation impact of which would never be observed. Such false violations might well be based on unreliable or unverified information sources. Reporting of these violations should invite punishment since they might be aimed toward falsely implicating others and being based upon non verifiable claims.

Rows categorize the reporting behavior of the players. We consider the cases of reporting of violations after they have occurred and of potential violations reported in advance, which may occur if suitable measures on enforcing the policies are not kept in place.

When a violation occurs, either s_i would report such a violation (having detected it) [Row 1] or it will go unreported. The case of non-reporting is further classified into two categories: i) Row 2 represents the scenario where s_i did not report and violation remained undetected (that is, no one else also reported it.) ii) Row 3 represents the scenario where s_i detected a violation but did not report it, while some other player detected as well as reported it – to establish such a case – we need to consider another payoff matrix as depicted in Fig. 2, which captures detection and reporting of such non reporting instances. The last row is meant to capture a potential violation or threat reported by s_i.

In the second table on Secondary Payoffs, first column - *True Secondary Violation* - represents that case, where player s_i detects a violation and also detects some other player(s) detecting the same violation though not reporting it. On the other hand, second column - False Secondary Violation - represents that scenario, where player s_i may act on the basis of a *false or fabricated scenario* and blame that such a scenario was witnessed by some other players but they did not report it. To elaborate these further, next we discuss each payoff entry in the tables.

In the following discussion, $\mathbf{PriTable}[\mathbf{m},\mathbf{n}]$ and $\mathbf{SecTable}$ $[\mathbf{m},\mathbf{n}]$ would denote the cell in m^{th} row and n^{th} column in Primary Payoff Table and Secondary Payoff Table respectively, where row/column indexing starts from 1.

3.1 Primary Payoff Table

The reporting behavior and corresponding payoffs for genuine (true) violations are represented in the first column and are discussed next:

$\mathbf{PriTable}[\mathbf{1},\mathbf{1}]$: Represents the scenario where player s_i detects violation vio_j and duly reports it and is rewarded with $R_{ij}(t)$. Actual value of the rewards could be determined based upon the characteristics of the violation vio_j, reporting delay etc. and can very well vary over time. Increase in the clearance level for subjects as defined in various mandatory access control models [2] can be considered as an example for such a reward.

In case majority of the players who detected and reported the violation also report that player s_i did not actually detect the violation but reported it only to get share in the reward, her reward could be withdrawn.

As discussed further in Section 7, in this paper we limit our scope to only reporting of the violations and do not consider the aspect of who committed the violation. However there exist a special case of self-reporting, where a user reports a violation committed by herself. Since positive rewards for such self-reporting might give rise to instances of deliberate violations with reporting by the users for their advantage, we suggest that in such cases the rewards should be kept to 0 (e.g., intangible positive feedbacks.)

$\mathbf{PriTable}[\mathbf{2},\mathbf{1}]$: Represents the scenario where a violation occurs but it is not reported by any of the associated players. This covers both the cases where violation was detected by some of the players but none of them reported it or when it remained undetected and hence was not reported. Notice that owing to the assumption of observability, even if a violation remains undetected, the consequences of the violation will still be observed in the future and thus it would get identified by the system administrators.

In such a case, each player pays a community price for it as denoted by -$CP_j(t)$. In case, if violations occur repeatedly, value of $CP_j(t)$ might also increase. Otherwise if the frequency of similar violations decreases over time, value of $CP_j(t)$ might also decrease.

Consider, for example, a source code is being changed or copied and transferred by some of the members of the project team and none of those who had knowledge of it reported it. Since its impact would be anyway felt at some stage later, all the associated players need to bear some loss for this.

Such a community price to be paid by each associated member is considered to be a critical component if such a model has to give rise to a dynamically evolving and increasingly secure system with collective responsibility. However in some cases, there might exist legal constraints, which limit explicit community punishments (e.g., Geneva Convention.) In such cases value of $CP_j(t)$ could be set to 0 (or an intangible punishment like negative group feedbacks.)

$\mathbf{PriTable}[\mathbf{3},\mathbf{1}]$: Represents the scenario where player s_i detects a violation but does not report it. We term it as *secondary violation* to distinguish it from the *primary violation*. Such a claim would be valid only when there exists some other player s_k, who also detects/witnesses the same violation vio_j and also detects that it has been witnessed by player s_i and s_k reports so. Note that s_k can also be a neutral monitoring device by which such a claim can be derived as well as verified. Therefore it is necessary to consider the cell $\mathbf{PriTable}[\mathbf{3},\mathbf{1}]$ for player s_i only in conjunction with the cell $\mathbf{SecTable}[\mathbf{1},\mathbf{1}]$ for some other player s_k.

$-P'_{ij}(t)$ denotes the price player s_i needs to pay for such non reporting of a violation. It can be argued that repeated occurrences of such non-reporting by a player must invite even harsher punishments, that is, $P'_{ij}(t)$ could be set as $P'_{ij}(t) = c.P'_{ij}(t-1)$, where c is some constant greater than one.

The difficult part in such a scenario is to validate the correctness of the claim reported by player s_k that player s_i witnessed the primary violation! In general it would require environment specific proofs (e.g. audio-video recordings etc.) However we believe that bare difficulty of proving such should not exclude such a scenario from the discussion.

$\mathbf{PriTable}[\mathbf{4},\mathbf{1}]$: Represents the scenario complimenting the scenarios considered in the earlier rows. Here player s_i *proactively reports* a *potential violation* and is therefore rewarded with $\Theta_{ij}(t)$.

Since a potential violation cannot be observed, it is assumed that it is logically possible to verify its truth by generating some hypothetical scenario where such violation would become possible. For example, for a newly created logical object, its owner subject/user might report potential access violations with the existing assess enforcement policies.

The reporting behavior and corresponding payoffs for false violations are represented in the second column and are discussed next:

$\mathbf{PriTable}[\mathbf{1},\mathbf{2}]$: Represents the scenario where player s_i falsely reports that violation vio_j has occurred in order to implicate other users, so need to be punished with $-P_{ij}(t)$. Again actual value of such punishment may depend upon the characteristics of the violation vio_j, past behavior of the player s_i etc. For example, in case, s_i is found to be repeatedly reporting false violations for implicating other users, associated punishments could increase correspondingly. Notice that every genuine violation is assumed to have some observable impact hence falsity of any such reported violation is verifiable (see the assumption of Observability).

$\mathbf{PriTable}[\mathbf{2},\mathbf{2}]$: Captures the scenario where violation vio_j has neither occurred nor has it been reported by s_i. It is associated with #, an undefined value.

$\mathbf{PriTable}[\mathbf{3},\mathbf{2}]$: This cell is meant to complete the table which captures an inherently false scenario where player s_i does not report a false primary violation (which of course cannot be detected by anyone else!) It is also associated with undefined value #.

$\mathbf{PriTable}[\mathbf{4},\mathbf{2}]$: Represents the scenario where player s_i reports a false potential violation. Similar to above, falsity of such a violation can be logically derived. We associate 0 value for the corresponding cell since it might not possible to prove that player s_i reported such false potential violation only with malicious intentions and incomplete information or a faulty analysis might be the basis for such a conclusion by s_i.

3.2 Secondary Payoff Table

The reporting behavior and corresponding payoffs for genuine (true) secondary violations are represented in the first column and are discussed next:

$\mathbf{SecTable}[\mathbf{1},\mathbf{1}]$: The first cell in the table represents the scenario where player s_i detects a violation and also detects that some other player(s) is(are) detecting the same violation but not reporting it.

This cell event can be true only if for the same player, event corresponding to $\mathbf{PriTable}[\mathbf{1},\mathbf{1}]$ is also true: it is a consistency check which states that secondary violation can be detected (and reported) only in conjunction with primary violation and not in isolation. The reward associated with this as represented by $r_{ij}(t)$.

$\mathbf{SecTable}[\mathbf{2},\mathbf{1}]$: Represents the scenario where a secondary violation occurs but it is not reported by any player. That means, there exists some user s_α, who detected the vio_j but did not report it. Also none of the other users having knowledge of this reported against s_α. Since it appears that in general independently establishing this is quite difficult and a secondary violation would not have serious negative impact on the whole community, we chose to give 0 as the value in this cell.

$\mathbf{SecTable}[\mathbf{3},\mathbf{1}]$: Represents the scenario where player s_i detects a secondary violation but does not report it. This is the case where it could be assumed from the context of the primary violation that with high probability several players must have detected such a violation but none of them reported it. In such a case, each player pays a community price for such complicity as denoted by -$cp_j(t)$.

This should be distinguished from the situation discussed in **PriTable[2,1]**, where a primary violation occurs but is not reported. The crucial difference is that there might exist certain situations, where primary violation would be by nature undetectable (e.g. when a violation occurs in isolation), therefore would go unreported as well - this is the case for **PriTable[2,1]**. On the other hand, there might also exist scenarios where primary violation must have been witnessed by or known to at least one player but was never reported (e.g. data manipulation on a shared document.), such cases are considered here.

Notice that we do not demand here that again some third player detects and reports such non-reporting of a secondary violation since we assume that it might not be possible in practice to continue to such an extent and such consideration might indeed lead to an indefinite regression.

$\mathbf{SecTable}[\mathbf{4},\mathbf{1}]$: Represents the scenario where player s_i reports on a possible violation vio_j and also that some other player(s) would detect the same violation but would not report it. This basically means s_i would be characterizing the potential behavior of certain other players who have greater probability of witnessing some violation vio_j. We associate some reward $\partial_{\mathbf{ij}}(\mathbf{t})$ with it.

The reporting behavior and corresponding payoffs for non existing false secondary violations are represented in the second column and are discussed next:

$\mathbf{SecTable}[\mathbf{1},\mathbf{2}]$: Represents the scenario where player s_i (falsely) reports that some other user(s) witnessed violation voi_j but did not report it so would to be punished with -$p_{ij}(t)$.

Notice that false secondary violation cannot be considered in isolation and need to be considered only in conjunction with either a true primary violation or in conjunction with a false primary violation. This is because if s_i has to report that some other user s_k witnesses violation voi_j, then s_i must also be reporting that violation vio_j occurred, which would imply that either $\mathbf{PriTable}[\mathbf{1},\mathbf{1}]$ or $\mathbf{PriTable}[\mathbf{1},\mathbf{2}]$ is also true for s_i.

$\mathbf{SecTable}[\mathbf{2},\mathbf{2}]$: Captures the scenario where no secondary violation has actually occurred and it has not been reported as well. # denotes an undefined value.

$\mathbf{SecTable}[\mathbf{3},\mathbf{2}]$: This cell captures an inherently false scenario where player s_i does not report a false secondary violation (which of course cannot be detected by anyone else!) It is also associated with undefined value **#.**

$\mathbf{SecTable}[\mathbf{4},\mathbf{2}]$: Represents the scenario where player s_i reports a potential false secondary violation. Such scenarios does not appear to have any serious relevance, hence we associate # with it.

3.3 Model Design Justification

A natural question which arises is the correctness and effectiveness of the model. This is important since collaboration by definition of the word cannot be guaranteed, in general.

Claim: *Assuming that there do not exist factors undermining the reporting behavior of individuals, under the proposed design of the payoff matrix model, at any point, individual gains from reporting true primary violations are always positive.*

Proof: The claim is based upon the following observation on the payoff matrix design: Suppose a player detects a primary violation. She would be faced with two choices – either she would proceed ahead and report the violation or she would not. In case of the former choice, she becomes entitled to receive the reward, which is a non negative value. Whereas, if she decides to remain silent on the violation, she is taking a risk of either loosing some value as a part of community price (provided no one else reports it either) or the risk of being punished for secondary violations in case there exist some other player who detected the violation and also detected that this player too had witnessed the same and the second player reports both of these violations.

So in case there do not exist factors, which counter these payoff matrix based rewards and punishments and motivate a player to remain silent on the violation, she would always be better off by reporting the violations detected. Ξ

This claim justifies the design of the model to be consistent with the motivation. The pay-off matrix model also discourages a user who understands the dynamics of the model not to falsely implicate another subject for a violation. Since the model only considers observable violations (Section 2.1), i.e., the violations impact of which can be observed, it rules out the possibility that a member of a group will be successful in alleging against a colleague by reporting a violation that has never occurred. Now it is possible that a subject may report an observed violation to be committed by someone who actually has not committed the violation. However, other users may also have detected the violation and reported the same. So by falsely implicating someone, the subject takes the risk of getting double punishment - one for not reporting the actual violation (violation committed by actual violator) and the other for reporting a false violation. Thus, by implicating a subject falsely for a violation, she always incurs the risk of getting punished.

In the pay-off matrix model, a subject is given reward for perception of a potential threat, but no punishment is given for wrong perception of a threat. There is a tread-off in giving and not giving punishment for reporting potential violation wrongly. If punishment is given for reporting an wrongly presumed potential threat (here we assume that the security administrators are able to determine if a reported potential violation is indeed possible to happen in future), then the subjects may not be willing to report a potential violation in the fear of getting it proved to be wrong, and getting punishment. On the other hand, there may be a many false positives among the reported potential violations. However, we have decided not to keep any punishment for reporting a wrong potential violation, as the pay-off matrix is defined for each group separately, and we assume that the members of a group have good judgments in identifying potential threats associated with the assets belonging to the group.

3.4 Economics of Payoff Tables

The idea that not reporting a violation would be treated as a violation could be a source of interesting social mechanisms. For example, it is possible that the system can apparently finance itself with no investment or revenue. This becomes possible when punishment for not reporting a detected primary violation, that is, -$P'_{ij}(t)$ is set at the same level as the reward for reporting the corresponding secondary violation, i.e., $r_{ij}(t) + P'_{ij}(t) = 0$ and rest all other parameters are set to 0. Here a user reporting secondary violation would be earning at the cost of that user who did not report the primary violation. Therefore, we need to introduce additional constraints on the parameters to avoid the scenarios which are economically infeasible.

For an instance of vio_j at time t, let $MaxLoss_j(t)$ be the maximum possible loss, which could have happened if the violation remained undetected and let $ActualLoss_j(t)$ be the actual loss even after the violation was duly reported. Therefore the effective gain from reporting can be estimated as

$$\Delta_j(t) = [MaxLoss_j(t) - ActualLoss_j(t)]$$

In case when violation goes unreported, $\Delta_j(t) = 0$ otherwise $\Delta_j(t) > 0$. Next, for all $vio_j \in Vio$ and $\forall t$, we define the following constraints:

$$\sum_{s_i \in S} [R_{ij}(t) + r_{ij}(t)] \leq \Delta_j(t) + \sum_{s_i} P'_{ij}(t) \qquad (1)$$

$$\sum_{s_i \in S} P'_{ij}(t) \leq \Delta_j(t) \qquad (2)$$

$$\sum_{s_i \in S} CP_{ij}(t) \leq MaxLoss_j(t) \qquad (3)$$

$$\sum_{s_i \in S} \Theta_{ij}(t) \leq MaxLoss_j(t) \qquad (4)$$

Eq.(1), guarantees that for every violation vio_j, total rewards received by all the users who reported the violation or reported secondary violation on it are no more than the effective gain by reporting it plus the punishments meted out to those users who did not report the violation even having detected it.

Eq.(2) guarantees that for every violation vio_j, total punishment for the secondary violation is not more than the effective gain which resulted by reporting it. Eq.(3) also similarly guarantees that total community punishment meted out to all the members is not more than the loss owing to the violation.

Eq.(4) guarantees that for every reported threat, which might occur, the total rewards received by all the reporting users is no more than then maximum loss possible owing to the violation (assuming that it also goes undetected).

4. IMPLEMENTATION ISSUES

In case of users as actual subjects, implementation of the collaborative monitoring model demands suitable framework for dismantling the information on the proposed payoff matrices to all the users as well as mechanisms for reporting the primary or secondary violations. Actual reporting structure for various policies and associated violations may differ based upon the organization type, type of the policy, user base, nature of the violations, and other associated environmental factors. For example, a user on a managerial position might report a violation, might receive a report from other users, and also could be an authority to enforce payoffs. Also associated payoff need to be decided in a time varying manner to render the system adaptive together with adequate confidentiality measures for protecting the identities of the reporting users.

4.1 Rewards and punishments: how to decide?

In general deciding appropriate rewards and punishments is critically dependent on the nature of the policy violations, their impact on the organization, ease of detecting them by the community members, and the nature of the groups associated with monitoring the policy violations etc. For example, with mandatory access control based security frameworks, employed for highly confidential assets (e.g. in military establishments), objects are differentiated according to their *sensitivity levels*, and the subjects are categorized based on their *trust levels*. Usually user accesses to different objects are limited according to their trust levels. There can be a number of schemes for defining the rewards and punishment criteria in terms of these levels. A simple scheme may be where a reward implies the increase in the trust level of a particular user, and punishment results into decrease in her trust level.

In reporting a violation, time is one of the important parameters. In general, the potential loss owing to a violation increases with increase in the *reporting delay*. So, reporting time may also play a role in deciding the reward for reporting a violation.

Let $\lambda(s)$ denote the trust level of subject s, and $\mu(o)$ denote the sensitivity level of an object o. The reward for reporting a violation of an access restriction on object o by subject s can be considered as follows:

$$\lambda(s) := \lambda(s) + f(\mu(o), r_t)$$

where $f(\mu(o), r_t)$ is any monotonically nondecreasing function of the sensitivity level o, and r_t, which denotes the reporting delay such that the value returned by the function increases with the increase in the value of $\mu(o)$ and decreases with the increase in the value of r_t.

A reward can alternately be defined in terms of reduction in loss owing to the timely reporting the violation. For example,

$$Reward(s, o) = \alpha.(MaxLoss - ActualLoss)$$

where α is some constant in the interval $[0..1]$.

Other parameters for rewards and punishments could also be defined accordingly for any given system set up.

We next discuss some generic guidelines based upon the studies on extrinsic motivation.

1. Reward induced behaviors in individuals tend to stop once the rewards are withdrawn (*overjustification effect* [13]). This fact places important constraints on deciding the rewards. For example, if rewards need to be withdrawn, it should be done gradually and also whenever intrinsic motivation is present, non tangible rewards (e.g., praise or recognition) should be preferred over tangible rewards.

2. Individuals evaluate the value of the rewards, which in turn determines their motivations for the tasks underlying the rewards, as compared to their current conditions (socio-economic status, responsibilities etc.) Hence rewards need to cater the satisfaction level of the individuals before they become effective.

3. Community price works as a negative reinforcement mechanism on the group level. Hence it would motivate people to monitor violations to avoid paying such price. Therefore for it to be effective, it is important that community prices are enforced strictly in the beginning though they should always be reduced as soon as reporting behavior has been adequately reinforced within the community.

4. Punishments for false reporting and secondary violations also work as negative enforcement for the individuals.

As noted for the assumption of policy-awareness in Section 2.1, sometimes users may not have the complete knowledge of a policy and therefore they might not be able to interpret correctly a witnessed scenario as an instance of a violation of the policy and therefore may fail to report it. Therefore it is suggested that for the first time, i.e., $t = 0$, if user does not report a witnessed violation, punishment for this (see **PriTable[3, 1]**) may be exempted if it turns out that the user was genuinely not aware that the witnessed scenario was a violation. Also it is possible that a user reports a false violation (see **PriTable[1, 2]**) because of the incomplete knowledge of the policy, that is, a user might presume a scenario as a violation though there is none, e.g., during an audit, people external to the group may be legally given some confidential information, however a user may presume it to be a violation and might falsely report this. In such cases also it is suggested that $P_{ij}(0) = 0$.

4.2 Correctness Properties

Let $r_{vio}(t)$ be *the number of violations per unit time distributed over t*, e.g. distribution on the number of violations per year. Similarly for the rate of reporting, let $r_{rep}(t)$ denote the *distribution of* the number of cases reported for true violations per unit time. Let $r_{false\_pri}(t)$ and $r_{false\_sec}(t)$ denote the distributions for the rate of occurrence of false primary and false secondary violations respectively.

Then probability distribution for the occurrence as well as reporting of a true violation can be approximated as

$$\frac{r_{rep}(t)}{r_{vio}(t)}$$

Since unlike the traditional security models, the proposed model is actually a monitoring model, we define the following robustness properties:

Probabilistic Robustness: A monitoring policy is termed as *probabilistically weakly robust* if over a course of time the rate of detections and reporting of true violations reaches the rate of actual violations and the rate of reporting of false violations decrease.

Formally,

$$\lim_{t \to \infty} \frac{r_{rep}(t)}{r_{vio}(t)} = 1$$
$$\lim_{t \to \infty} r_{false\_pri}(t) = 0$$
$$\lim_{t \to \infty} r_{false\_sec}(t) = 0$$

Probabilistic Strong Robustness: A monitoring policy is called *probabilistically strongly robust* if over a course of time the rate of access restriction violations steadily reduces. Formally,

$$\lim_{t \to \infty} r_{vio}(t) = 0$$

5. PROBABILISTIC MODEL FOR PARAMETER ESTIMATION

Dynamics of collaborative monitoring depends on various factors. Firstly, not all policy violations are equally likely to be detected. Moreover, if a user detects a violation, whether she would actually report the violation or not depends on different factors, for example, the rewards she would get for reporting, the punishment that she would invite if she does not report, and also any hidden incentives associated with not reporting the violation. Therefore, we model the system as a probabilistic system, more precisely as a basic Markov Decision Process (MDP without rewards), to estimate certain reporting probabilities and experimentally demonstrate how model checking based approach can help an administrator determine different parameters in the Payoff Matrix. In practice, the model needs to be initialized using Bayesian probabilistic estimates by the administrators using historical data or other associated analysis to support these estimates. However as we discuss later, some of these probabilities get refined iteratively as new data becomes available over time.

Let p_{det_j} be the probability that a violation vio_j could be detected by any subject, which indicates the inherent difficulty in detecting the violation. Similarly let $p_{det\_sec_{ij}}$ denote the probability that subject s_i detects a secondary violation by any other subject on violation vio_j. The probability p^{pri}_{ij} denotes that the subject $s_i \in S$ will report a primary violation vio_j. Similarly the probability p^{sec}_{ij} denotes that the subject s_i will report a secondary violation on vio_j.

We next define a *motivation index*, m_{ij} for a subject s_i to report a violation vio_j. Motivation index is a measure of the motivation a subject has for reporting a violation. The motivation index can be considered to be determined by the following factors:

1. Individual gain from the reward.
2. Fear of Community price or punishment for secondary violation.
3. A number of factors that collectively can act as a deterrent for reporting the violation. For example, personal relationships with the violators or potential collusion, incentives offered by the violators, possible altruism, or delusional, consistent irrationality.

In general quantitative measures for these factors are situational, however we may consider the following measure for defining m_{ij}:

$$m_{ij} = |T^P_{ij}[1,1]| + \max\{|T^P_{ij}[2,1]|, |T^P_{ij}[3,1]|\} - \Omega_j$$

where $T^P_{ij}[1,1]$ is the reward, s_i would gain for reporting true violation vio_j, $T^P_{ij}[2,1]$ is the corresponding community price if none of the subjects detecting the violation report it, and $T^P_{ij}[3,1]$ is the punishment for the secondary violation, that is, the loss s_i would have in case she does not report the violation but in turn some other subject reports against him for doing so. Ω_j indicates the effect of the factors that collectively can act as a deterrent for reporting the violation (point 3 above). For simplicity, it is defined as a fraction $\delta \in [0,1]$ of the $MaxLoss_j$, which is the maximum loss caused by the violation:

$$\Omega_j = \delta * MaxLoss_j$$

In this definition we assume that the factors which would work against reporting a violation could be indirectly considered as being related with the 'share' in the gain subject s_i may have by not reporting the violation. Under such formulation, a probabilistically weakly robust monitoring policy would require that violations by a group of users should be very difficult so that most of the users in the group other than the violator himself may become potential witnesses.

As a mathematical simplification, we also enforce that $m_{ij} \le M$, where M is some large positive constant upper bounding m_{ij}. We further assume that the probability of reporting a violation by s_i is related to m_{ij} as follows:

$$p^{pri}_{ji} = \begin{cases} \frac{m_{ij}}{M} & \text{if } m_{ij} \ge 0 \\ 0 & \text{if } m_{ij} \le 0 \end{cases} \qquad (5)$$

Next consider that the number of subjects detecting a violation vio_j follows a probability distribution with mean k. So that, at any instance, for vio_j a subset of subjects $S_j = \{s_{i_1}, s_{i_2}, \ldots, s_{i_k}\} \subseteq S$ detect the violation. Note that the probability that $S_j = \emptyset$ at any instance is $1 - p_{det_j}$. Based upon these, we can estimate the following:

The probability with which the violation vio_j will be reported is

$$1 - \prod_{s_l \in S_j} (1 - p^{pri}_{lj})$$

For each player $s_l \in S_j$, let us consider the subset of the players who notice s_l detecting vio_j as $Y_l = \{s_{l_1}, s_{l_2}, \ldots, s_{l_r}\} \subseteq S_j$. The probability that at least one of the players from Y_l would report the secondary violation against s_l can be estimated as

$$\xi_{lj} = 1 - \prod_{s_t \neq s_l \in Y_l} (1 - p^{sec}_{tj})$$

In general, we can chose Y_l either nondeterministically or probabilistically. Under nondeterministic choice the probability that other players would report the secondary violation against s_l can be estimated as

$$\xi^{nd}_{lj} = 1 - \sum_{Y_l \in 2^{S_j}} [\prod_{s_t \neq s_l \in Y_l} (1 - p^{sec}_{tj})]$$

Probabilistic choice on the other hand demands a probability measure

$$\mathbf{Dist}_j : S_j \times 2^{S_j} \mapsto [0,1]$$

For $s_l \in S_j$ and $Y_l \subseteq S_j$, $\mathbf{Dist}_j(s_l, Y_l)$ is the probability that all the subjects in Y_l notice s_l detecting violation of vio_j. We require that

$$\sum_{s_l \in S_j \wedge Y_l \subseteq S_j} \mathbf{Dist}_j(s_l, Y_l) = 1$$

Then the probability that other players would report the secondary violation against s_l can be estimated as

$$\xi^{prob}_{lj} = 1 - \sum_{Y_l \in 2^{S_j}} [\mathbf{Dist}_j(s_l, Y_l) * \prod_{s_t \neq s_i \in Y_l} (1 - p^{sec}_{tj})]$$

Based upon the claim above, we can estimate the probability with which a secondary violation will be reported as

$$\sum_{s_l \in S_j} (1 - p_{lj}^{pri}) * \xi_{lj}$$

Let us next estimate the probabilities corresponding to possible reporting behaviors by a subject on a violation vio_j:

Consider the case where a subject $s_i \in S_j$ reports both primary as well as secondary violations. Using $\mathbf{Dist_j}$, we can estimate the probability that s_i would detect a secondary violation as

$$\xi_i^{sec} = \sum_{s_l \neq s_i \in S_j \wedge s_i \in Y_l \subseteq S_j} \mathbf{Dist}_j(s_l, Y_l)$$

Therefore the probability that s_i would report both primary as well as secondary violations for vio_j is

$$p_{ij}^{pri} * p_{ij}^{sec} * \xi_i^{sec}$$

Consider the case where s_i reports primary violation but not the secondary violations, even though she may detect them. The probability of such occurrence depends on the probability that s_i would report primary violation on vio_j, probability that s_i would detect a secondary violation, and the probability that s_i would not report this secondary violations. As the case above, the total probability for this case is

$$p_{ij}^{pri} * \xi_i^{sec} * (1 - p_{ij}^{sec})$$

On the other hand, consider the case where s_i reports primary violation but not the secondary violations since s_i did not actually detect that. The probability of such occurrence is

$$p_{ij}^{pri} * (1 - \xi_i^{sec})$$

Finally consider the case where s_i does not reports primary violation and so by design of the model would not report secondary violations too. The probability of such occurrence is

$$(1 - p_{ij}^{pri})$$

6. EXPERIMENTAL ANALYSIS

For experimental analysis of the above system model we use PRISM model checker [22] and express desired properties in terms of PCTL (Probabilistic Computation Tree Logic) [16]. PRISM is a tool for formal modeling and analysis of systems which exhibit probabilistic behavior including MDPs and provides support for automated analysis of a wide range of quantitative properties of these models.

6.1 Modeling with PRISM

For any model-checking activity the behavior of the underlying system is abstracted as transition system. In order to construct and analyze a model with PRISM, it needs to be be specified in the PRISM language, a simple, state-based language, based on the Reactive Modules formalism of Alur and Henzinger [1].

The fundamental components of a PRISM model are modules. A model is composed of a number of modules which interact with each other. A module contains a number of local variables. The values of these variables at any given time constitute the state of the module. The global state of the whole model is determined by the local state of all modules. The behavior of each module is described by a set of commands. A command is of the following form:

$$[\,]\, condition \Rightarrow$$
$$p_1 : update_1 + p_2 : update_2 + \ldots + p_n : update_n;$$

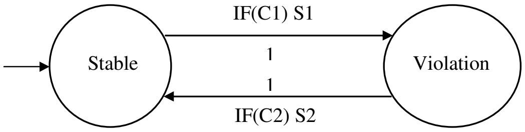
The $condition$ acts as a guard which is a predicate over all the variables in the model (including those belonging to other modules). Each $update_i$ describes a transition which the module can make with probability p_i if the $condition$ is true . A transition is specified by giving new values to the variables in the module, possibly as a function of other variables.

The PRISM model in this work consists of two kinds of modules: A module for the 'environment' considered to be generating violations and a module for a subject detecting either primary violation or both primary and secondary violations. These modules are discussed next:

6.2 Environment Module

We capture the occurrence of a violation in an *environment* module in the PRISM model as depicted in Figure3. The violations are assumed to be occurring independent of each other. Therefore, we consider only one violation in our experiments and study the consequences related to it. We will omit the subscripts for the violation in the following discussion.

States of environment module are denoted by $state\_env$ variable and the states of subject s_i are represented using $state\_subi$. A violation may occur only when the system is in a *stable* state. When all the subjects complete their reporting activities related to the violation, the system again returns to the *stable* state. The state transition diagram of the model of environment is shown in Figure 3.



C1 :: state_sub1 = stable ∧ state_sub2 = stable ∧ ... ∧ state_subn = stable;
C2 :: state_sub1 = end ∧ state_sub2 = end ∧ ... ∧ state_subn = end;
S1:: state_env := violation; S2:: state_env := stable;

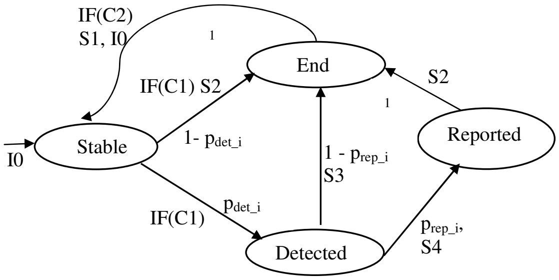
Figure 3: State Transition diagram for the Environment Module

6.3 Module for a Subject Detecting only Primary Violations

Figure 4 depicts the transition diagram for a subject. Referring to the figure, a subject stays in a *stable* state when no violation occurs. When a violation occurs, as captured by the condition $C1$ in the Figure 4, a subject may or may not detect the violation based on detection probability. Therefore, from the *stable* state, the subject can go to state *detected* with probability p_{det} and to state *end* with probability $1 - p_{det}$. If the subject is in *detected* state, it can either report the violation with its reporting probability p_{rep} and transit to *reported* state, or it may not report the violation with probability $1 - p_{rep}$ and in turn may transit to the *end* state. After reporting the violation the subject finally moves to *end* state. When all subjects are in their *end* states and there is no more activities from the subjects regarding the violations, environment module can then move to its *stable* state. When environment is in *stabl*e state after a violation, all the subjects also move to their *stable* states.

We use a flag to distinguish two different possible behaviors of a subject after detecting a violation. In *stable* state, the flag is set to 0. If a subject reports the violation, its flag is set to 1 on taking a transition to the state *reported.* Otherwise if the subject does not report the violation after detecting it, its flag is set to 2. When the subject moves from *end* state to the *stable* state, the flag is set to

0. This flag is used in writing PCTL properties and for modeling secondary violation, discussed next.



C1 :: state_env = violation; C2 :: state_env = stable; I0:: $flag_i$:= 0;
S1:: $state\_sub_i$:= stable; S2:: $state\_sub_i$:= end; S3:: $flag_i$:= 2;
S4::$flag_i$:= 1;

Figure 4: State Transition diagram for a Subject s_i Detecting only Primary Violations

6.4 Module for a Subject Detecting both Primary and Secondary Violations

As depicted in the Figure 5, the module for a subject reporting only the primary violations can be extended to capture the activity of the subject related to secondary violations. The primary condition of detecting and reporting a secondary violation is that the subject has to report the corresponding primary violation also. So in the model of a subject for primary violation if the subject is in *reported* state, the subject may detect secondary violation by the other subject. We shall illustrate the model for two subject system. From the *reported* state, the subject may detect a secondary violation with probability $p_{det_{sec}}$ and may move to *sec_vio_detected* state with probability p_{det_sec} and *end* state with probability $1 - p_{det\_sec}$. From *sec_vio_detected* state, the subject may move to *sec_vio_reported* with probability p_{rep_sec} or may move to the *end* state with probability $1 - p_{rep\_sec}$. If a subject reports a secondary violation after detecting it, its flag is set to 3, otherwise the flag is set to 4. In Figure 5 $flag_i$ denotes the flag for the subject for which we are considering the model and $flag_j$ corresponds to the other subject.

6.5 The Combined System

The combined system can be represented as

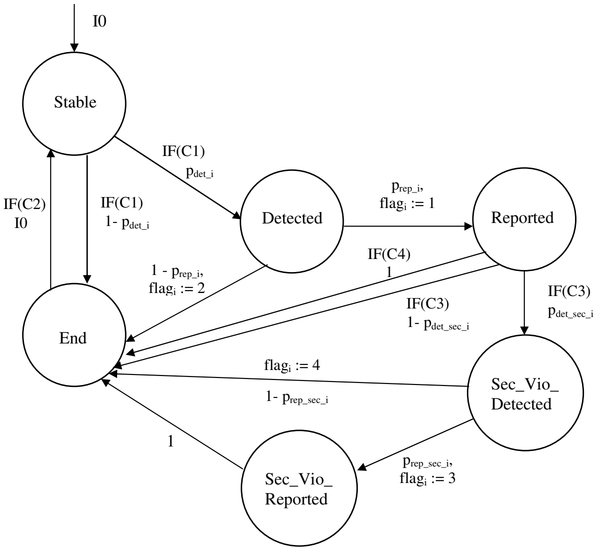
$$Sys:\ \{\theta\}[Env||Sub_1||\ldots||Sub_n]$$

Where *Env* denotes the environment module used for generating violations. $Sub_1 \ldots Sub_n$ model the behavior of the subjects s_1, $s_2, \ldots, s_n$. θ specifies the initial values of variables.

6.6 Properties of Interest

In order to estimate the desired probabilities, we specify properties in PCTL. Since nondeterminism is involved, PRISM calculates the maximum and the minimum probability of a property being satisfied considering the best and worst cases after resolving all nondeterminism. For primary violation, we are interested in estimating the probability of a violation to be reported by at least one subject. The following PCTL property specifies this:

$$P_{min} =?[(s = 1) \Rightarrow \mathbf{F}\left(\text{`}Report'\&(s = 0)\right)]$$



C1:: state_env = violation; C2 :: state_env = stable;
C3:: $flag_j$ = 2; C4 :: $flag_j \neq 2$; I0:: $flag_i$:= 0

Figure 5: State Transition diagram for a Subject Detecting Primary and Secondary Violations

Where s denotes the state of the *environment* module. Predicate $(s = 0)$ checks if the environment module is in *stable* state and $(s = 1)$ checks whether environment module is in *violated* state. $f_1, f_2, \ldots, f_n$ denote the flag associated with subjects s_1, s_2, …, s_n. When value of a flag is 1, it indicates that the corresponding subject has reports the violation. Therefore, the predicate '$Report'$ is defined as $Report \equiv ((f_1 = 1)|(f_2 = 1)|\ldots|(f_n = 1))$ denoting that at least one of the subject detects and reports the violation. **F** is the 'eventually' or 'in the future' operator. Finally, the property states that P_{min} *is the minimum probability that if a violation occurs, it would be eventually reported by at least one subject.*

The next property specifies the probability of reporting a secondary violation by subject s_1 against subject s_2:

$$P_{min} =?[(f_2 = 2) \Rightarrow \mathbf{F}((f_1 = 4)\&(s = 0))]$$

where $(f_2 = 2)$ denotes that subject s_2 has detected but not reported the primary violation, thus committed a secondary violation. $(f_1 = 4)$ denotes that subject s_1 has reported this secondary violation.

6.7 Experimental Results

Experimental evaluation provides insights as to how different parameters such as detection probability, motivation index, and number of subjects contribute to reporting probability of a violation. In discuss here the experiments on primary violation. In the experiments, one of the three parameters was kept constant and remaining two parameters were varied to determine the effect of the changes in these parameters on the reporting probability. A C program was developed to automate the process of generating these PRISM models with these parameters and a property file containing the properties discussed before. Finally, the required probability is extracted from the output file populated by the PRISM.

Figure 6 shows the variation of reporting probability with changes in the number of subjects and motivation index for detection probability = 0.5. Administrator can get useful insight from this kind of experiment. If an administrator can determine the detection probability for a policy violation from her experience, and if the number of associated subjects is also known, the required value of the motivation index can be assessed to achieve a particular reporting probability for the violation. This knowledge would in turn be used to determine the values for different entries in the payoff matrix for a subject-violation pair corresponding to the evaluated motivation

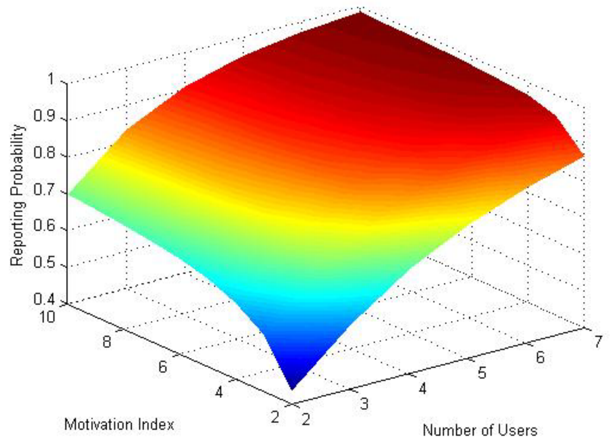


Figure 6: The variation of reporting probability with changes in the number of subjects and motivation index for detection probability = 0.5

index and associated reporting probability.

Figure 7 shows the variation of reporting probability with changes in the detection probability and the motivation index for number of users = 5. This is useful in the scenarios where a group of subjects are associated with an asset for which different violations are possible, and detection probabilities for these violations are also different. Figure 7 will give an administrator useful information about the motivation index for different violations for the same group of subjects.

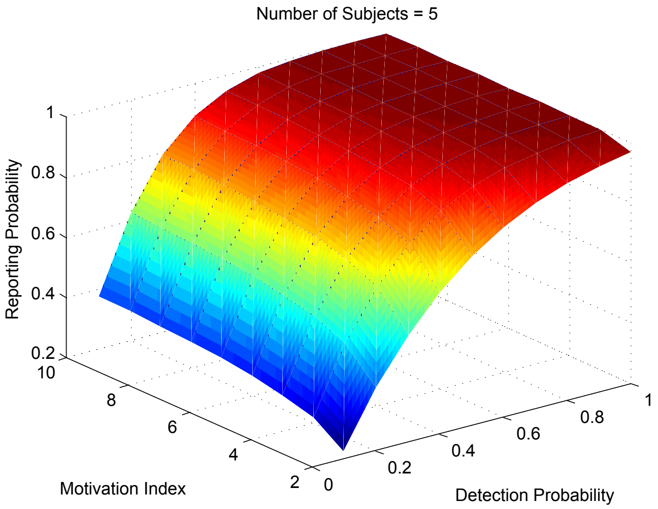


Figure 7: The variation of reporting probability with the changes in the detection probability and motivation index for number of users = 5

6.8 A Note on Determining and Updating Detection Probabilities

While deploying the collaborative monitoring system, the administrator has to determine the detection probability for a violation from her experience and historical analysis of the violations in past. This approach may get subjective, and sometimes the estimated values might be far away from the correct values. However to deploy the collaborative monitoring system, it is required to start with some values for detection probability. With some enhancement in the analysis, it is possible to have a better estimate of detection probability for some violation using the values for the total number of violations, the number of primary violations reported by a subject, and the number of secondary violation reported against the subject over a period of time.

Let us assume that the time period which is considered for estimating the detection probability of a subject s_i for violation vio_j is d time units. Let, in these d time instances, number of primary violation reported against violation vio_j is N. Also, the number of primary violation reported by subject s_i is n_p and the number of secondary violation reported against subject s_i is n_s. So, if the actual detection probability of subject s_i for violation v_j is p_{det_actual}, then

$$p_{det\_actual_{ij}} \geq \frac{n_p + n_s}{N}$$

So, the administrator now has a new estimate for the detection probability of a subject for a violation. Let us denote this new detection probability of subject s_i for violation vio_j as:

$$p_{det\_new_{ij}} = \frac{n_p + n_s}{N}$$

Now the administrator needs to run the experiment again to get estimate of new reporting probability, or to estimate new motivation index for achieving the previous reporting probability. Note that the detection probabilities now may be different for different subjects. Though in our previous discussion, we have considered same detection probability for all the subjects, the model can be easily enhanced for different detection probability for different subjects, as the models for individual subjects are independent from each other.

7. CHALLENGES FOR COLLABORATIVE SECURITY

Success of collaborative security entails a member in an organization to report violations against her known colleagues. However, a member of a group, where the members enjoy camaraderie, may not always be willing to report a violation (especially in the beginning) committed by a colleague, assuming that her action could bring punishment to her fellow member(s) and may in turn endanger her own social isolation in the group. This brings the challenge of setting the reward and punishment policies suitably so that they can effectively counteract any reason that thwarts a member in a group to report a violation. Selecting the types of reward is yet another challenge. As the research in the area of motivational psychology demonstrates, individuals differ in their preferences for rewards - some people may be motivated by monetary rewards, while some others may be more interested in recognition, while others may aspire for career advancement, and on certain scenarios individuals may not even prefer having any explicit (or tangible) rewards and just an enabling framework for reporting and consequent reduction in the potential loss to the organization itself would be sufficient for them. Indeed a reward is an ‘abstract token’ working as a psychological catalyst for the motivation of the users. Similarly punishment is also an ‘abstract token’ to demotivate users for not reporting the observed violations. The challenge present here is that since the same reward may not motivate all the members equally, this in turn might affect their reporting behaviors also.

Another challenge for collaborative security is setting up adequate regulatory controls so that the fundamental privacy rights of the members of the organization can be preserved. Maintaining anonymity of a person who reports a violation is often critical, as most people do not want to be perceived as whistle-blowers (or 'informers') by their colleagues and face social isolation. Sometimes this might make it difficult to punish the offender as punishing an offender would require an witness in most of the cases. In this respect, punishment against secondary violations should be beneficial as that would motivate all the witnesses of a violation to report it. Also studies on *social conformity* [5, Chap. 7],[24] (also known as 'normative conformity' [7]) demonstrate that often individuals fear to get into those acts alone which are not so well accepted yet by others in the group (e.g., reporting against peers) and prefer acting only in groups. On the other hand, studies on *social loafing* [20] and *bystander effect* [36] demonstrate that often individuals also feel a 'diffusion of responsibility' [23] when they are part of larger groups unless they are assigned clearly defined individual responsibilities. These conflicting factors demand that strong and effective initiatives to enforce collaborative security need to be adopted, which could counter the conformity biases, minimize negative peer influences [12], and motivate individuals to take leading roles in the process according to the prevailing socio-cultural environment in the groups/organization. In this respect consideration of secondary violations again could work as effective control for the individuals and could help reducing the bystander effect.

Next let us consider the assumption of policy consistency and completeness as described in Section 2.1. In practice, it might be difficult to meet these requirements always. However, it turns out that these are critical for the success of the collaborative monitoring framework we discuss in this paper. To see this, consider the case of contradictions in policy definition. If a policy is not contradiction free, there would exist scenarios where it will be possible to interpret these scenarios both as a violation of the policy as well as in accordance with the policy. This may also imply an ambiguity in the interpretation of its observable impact on the system (ref. 'axiom of observability' in Section 2.1). Therefore if a user reports it as a violation, she might expect due rewards, while system administrators might conclude otherwise and deny that reward to her. This in turn would negatively impact the motivation of the users to further report the violations in future. On the other hand, if a user concludes that the witnessed scenario does not constitute as a violation, she might not report on it, however some other users or the system administrators might interpret it differently and might impose punishment on her for non-reporting - which would again result in reduced motivation for further reporting. Therefore, it is absolutely necessary that policies are free from contradictions and are mutually consistent. Wherever it is not possible, a balanced approach towards deciding rewards and punishments might possibly help in dealing with it.

In the collaborative monitoring model, as the users are involved in detecting and reporting a violation, it may give an impression that the reported violations may have many false positives - a member in a group may allege falsely against some other member in the group to gain personal advantage or to cause harm to the colleague. However, as the model only considers observable violations, i.e., the violations impact of which can be observed, this would rule out the possibility that the member will be successful in alleging against a colleague by reporting a violation that has never occurred. It is, though, possible that a user may report an observable violation to be committed by someone who actually has not. However, the payoff structure as stipulated in this paper, is limited to the reporting of the 'occurrence of the violation' and leaves it outside the scope as to 'who actually committed the violation?' and also 'who has been blamed for it by the reporting users?', though both these aspects are often tightly coupled. The primary reason for such a scope limitation is that the truth of who actually committed the violation can only be established through forensic investigation and even after using sophisticated and reliable mechanisms it is possible that non-violators are suspected as violators owing to various confounding circumstantial reasons. Therefore, we do not expect that reporting users would be able to always suspect/identify correctly the actual violators and therefore do not stipulate any reward (or punishment) for merely reporting (correctly or falsely) who the violator(s) was(were) and leave such aspects to the pre-existing security enforcement mechanism. However, effectiveness of such post reporting investigations and enforcement of the punishment to the violators (or those who are proved to be framing non-violators knowingly) could also have an impact on the future reporting behavior of the users (especially if the existing law and enforcement system is not so effective) since these would act as pointers for their justification towards determining the overall utility and effectiveness of the reporting actions. Indeed, for a concrete realization of the presented framework, it is important to decide who verifies the reported violation and who approves the payouts, which would be determined by the existing corporate governance structure and policies of the organization.

Finally, a big computational challenge for the proposed probabilistic model for parameter estimation (Section 6) is how to make it scalable to meet the needs of large organizations. The success of the payoff model largely depends on estimating the values of rewards and punishments properly. The experimental setup presented for estimating different parameters of a payoff matrix does not appear to scale well. However, our hope is that with the advancement of tools and techniques in the field of formal methods, we will be able to reason about larger systems in near future. Moreover, the collaborative security model is modular in the sense that the model considers individual groups separately. Therefore, we do not need to always think about scaling the framework up to the whole organization level, the framework needs to be capable of scaling up to the group levels. The size of different groups based on different context may be of varying size though.

8. CONCLUSION AND POINTERS FOR FUTURE WORK

This paper presents a principled approach to one of the many little-studied aspects of computer security which relate to human behavior. Existing security frameworks often differentiate user base of assets from the security administrators who design and enforce the security policies. However, in many aspects of security, it is the user who is best suited to detect and prevent violations which currently lie beyond the scope of available security enforcement mechanisms. This is especially true for the violations on context dependent logical resources, e.g., some data, significance of which is realized only when considered with respect to specific contexts. Such resources are often vulnerable to 'semantic manipulations' and securing them using automated monitoring is either not feasible or would be very expensive.

In this respect, involving users, who usually have strong analytic ability to detect violations and threats but are not primarily responsible for security, can be quite advantageous. In this work, we have presented a generic reinforcement framework for enabling monitoring and detection of (potential) violations by these users. The probabilistic analysis, associated state-transition model for PRISM, and the experiments demonstrate how specific parameters can be estimated for determining the reward-punishment based policies for collaborative monitoring.

In our experiments, we implicitly assumed that individual motivation alone can determine the likelihood of reporting of a violation by a user as modeled by Eq 5, which is still a high level abstraction and leaves the scope of further work in this direction. The objective of this work would be to relate human behavior with intrinsic or extrinsic rewards and losses in a more detailed manner. Work in the direction of human behavior modeling [32] would contribute concretely toward this goal. Further analysis would require modeling the reporting behavior of users for secondary violations in a general setting involving n players. This may in turn enable a derivation of closed form solutions for optimal estimates of parameters in the pay off model for various security scenarios. In practical situations distributed and collaborative strategies may also be required for such estimates. Also, we need to model more realistically the external environmental factors which could control the reporting behavior of the users. Such modeling would give rise to extended game theoretic model for the overall system, equilibrium of which may shed further practical insights on designing policies for collaborative security.

The nature of emergent network effects [21, 25] under the proposed reward-punishment based reinforcement framework is yet another direction for future investigations. Currently when a new user joins a collaborative network/group, the detection probability for the violation(s) against the policies associated with the group would increase under the normal assumptions. This would in turn reduce the chances of other users fined for community price. This essentially induces '+ve' network effect. Also on the other side, new members increase the overall probability of detecting secondary violations in the group, which would also in turn may have an effect on other users reporting the detected violations. These network effects become even more important in the presence other environmental factors e.g., community structure [28, 27]. Development of analytical and/or experimental models to study the emergent macro level properties of the system e.g., plateau and reverse effects, small world effects, network resilience, and phase transitions [26] may potentially help in this direction.

User driven policy synthesis is an important aspect of collaborative security approach. In this work, we only consider users reporting violations and leave it upon the existing policy synthesis machinery to use these reported violations for defining new policies or refining existing ones. An important problem to be addressed when users are allowed to add new policies is the consistency checking and completeness analysis.

It is also interesting to analyze how learning could be enabled in the system. One way to introduce learning in the system is for deciding optimum values for the payoff parameters e.g., rewards and punishments. Modifying the system to learn about inconsistencies could also be considered as another direction for future work.

The framework could be extended with *decoy violations*, which could be used to test the possible user and group response behaviors for detection and reporting. Also if suitably designed, decoy violations could induce 'decoy-effect' (or 'asymmetric dominance effect') [17] in the system motivating users to prefer to reporting rather than not. Though decoy violations appear to have important role to play in the beginning, we need to understand if they also necessarily hold long term effects in the context of collaborative security. Also, the degree of correspondence between decoy violations and the Prisoner's Dilemma [31], in particular, iterated Prisoner's Dilemma needs to be further explored.

9. ACKNOWLEDGMENTS

The authors would like to thank anonymous referees and attendees of NSPW for their valuable feedback. Special thanks for the interest and efforts by the shepherds Deborah Frincke (before the workshop) and Mary Laura Corriss (after the workshop). Scribed notes by Matt Bishop and Mary Ellen Zurko have been very helpful in improving the paper post workshop. Finally, the registration and travel support given by the NSPW organizers to one of the authors was critical to participate and present the paper. We also thank Rupak Majumdar for his valuable feedback on the paper.

10. REFERENCES

[1] R. Alur and T. Henzinger. Reactive modules. *Formal Methods in System Design*, 15(1):7 – 48, 1999.

[2] M. Bishop. *Computer Security: Art and Science*. Addison Wesley, 1st edition, 2003.

[3] M. Bishop, S. Engle, S. Peisert, S. Whalen, and C. Gates. We have met the enemy and he is us. In *Proceedings of the 2008 New Security Paradigms Workshop*, Lake Tahoe, CA, 2008.

[4] M. Bishop and C. Gates. Defining the insider threat. In *Proceedings of the 2008 Cyber Security and Information Infrastructure Research Workshop*, Oak Ridge, TN, 2008.

[5] K. S. Bordens and I. A. Horowitz. *Social Psychology*. Lawrence Erlbaum, second edition, 2001.

[6] D. Cappelli, A. Moore, T. J. Shimeall, and R. Trzeciak. Common sense guide to prevention and detection of insider threats. Technical Report 2.1, Carnegie Mellon University, CyLab, 2006.

[7] R. Cialdini and N. Goldstein. Social Influence: Compliance and Conformity. *Annu. Rev. Psychol*, 55:591–621, 2004.

[8] E. Cole and S. Ring. *Insider Threat: Protecting the Enterprise from Sabotage, Spying, and Theft*. Syngress Press, 2006.

[9] R. Daniel. Monetary incentives, what are they good for? *Journal of Economic Methodology*, 12(2):265–276, 2005.

[10] S. di Vimercati and P. Samarati. Access control in federated systems. In *Proceedings of the 1996 workshop on New security paradigms*, pages 87–99. ACM New York, NY, USA, 1996.

[11] F. Ernst and F. Armin. *Psychological Foundations of Incentives*. IEW - Working Papers, Institute for Empirical Research in Economics - IEW, 1st edition, 1995.

[12] W. Felps, T. Mitchell, and E. Byington. How, when and why bad apples spoil the barrel: negative group members and dysfunctional groups. *Research in Organizational Behavior*, 27:181 – 230, 2006.

[13] B. Greene, D. abd Sternberg and M. R. Lepper. Overjustification in a token economy. *Journal of Personality and Social Psychology*, 34:1219–1234, 1976.

[14] S. Greenwald. A new security policy for distributed resource management and access control. In *Proceedings of the 1996 workshop on New security paradigms*, pages 74–86. ACM New York, NY, USA, 1996.

[15] M. T. Group. Do you really know what your programmers are doing? White Paper, `http://www.mintaka.com/whitepaper/WhitePaper-Security.pdf`, last accessed on April 04, 2009, Oct. 2008.

[16] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, 1994.

[17] J. Huber, J. Payne, and C. Puto. Adding asymmetrically dominated alternatives: Violations of regularity and the similarity hypothesis. *Journal of Consumer Research*, 9(1):90, 1982.

[18] A. Iyer and H. Q. Ngo. Towards a theory of insider threat assessment. In *Proceedings of the 2005 International Conference on Dependable Systems and Networks*, pages 108–117, Washington, DC, USA, 2005. IEEE Computer Society.

[19] M. Kabay. Using Social Psychology to Implement Security Policies. *Computer Security Handbook*, pages 35.1–35.22, 2002.

[20] S. J. Karau and K. D. Williams. Social loafing: A meta-analytic review and theoretical integration. *Journal of Personality and Social Psychology*, 65:681 – 706, 1993.

[21] K. Kilkki and M. Kalervo. Kk-law for group forming services. Presented at XVth International Symposium on Services and Local Access, Mar. 2004.

[22] M. Kwiatkowska, G. Norman, and D. Parker. Prism: Probabilistic symbolic model checker. In *Proceedings of PAPM/PROBMIV'01 Tools Session*, pages 7–12, 2001.

[23] B. Latene. The psychology of social impacts. *American Psychologist*, 36:343–356, 1981.

[24] R. Martin and M. Hewstone. Conformity and independence in groups: Majorities and minorities. *Blackwell Handbook of Social Psychology (Group Processes)*, pages 209–234, 2001.

[25] B. Metcalfe. Metcalfe's law: A network becomes more valuable as it reaches more users. *Infoworld*, (17), 1995.

[26] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

[27] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of National Academy of Science USA*, 103:8577, 2006.

[28] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.

[29] A. Odlyzko and B. Tilly. A refutation of metcalfe's law and a better estimate for the value of networks and network interconnections. `http://www.dtc.umn.edu/~odlyzko/doc/metcalfe.pdf`, last accessed on April 04, 2009.

[30] H. Petri. *Motivation: Theory, Research and Application*. Wadsworth Publishing, 5th edition, 2003.

[31] W. Poundstone. *Prisoner's Dilemma*. Doubleday New York, NY, USA, 1993.

[32] A. J. Puleo. Mitigating insider threat using human behavior influence models. Master's thesis, Air Force Institute of Technology, School of Engineering and Management, 2006.

[33] M. R. Randazzo, M. Keeney, E. Kowalski, D. Cappelli, and A. Moore. Insider threat study: Illicit cyber activity in the banking and finance sector. Technical Report CMU/SEI-2004-TR-021, Software Engineering Institute, Carnegie Mellon University, 2005, 2005.

[34] J. B. Rotter. Generalized expectancies for internal versus external control of reinforcement. *Psychological Monographs*, 80(1):1–28, 1966.

[35] J. B. Rotter. Internal versus external controls of reinforcement. *American Psychologist*, 45:489–193, 1990.

[36] M. Rowe, L. Wilcox, and H. Gadlin. Dealing with – or reporting – “unacceptable" behavior (with additional thoughts about the “bystander effect"). *Computers & Security*, 21(6):526 – 531, October 2002.

[37] C. Sansone and J. M. Harackiewicz. *Intrinsic and Extrinsic Motivation: The Search for Optimal Motivation and Performance, 1st edition*. Academic Press, 1st edition, 2000.

[38] E. E. Schultz. A framework for understanding and predicting insider attacks. *Computers & Security*, 21(6):526 – 531, October 2002.