Applying Norms and Sanctions to Promote Cybersecurity Hygiene

Submitted Version

Shubham Goyal
Amazon
Seattle, WA
gsshubha@ncsu.edu

Nirav Ajmeri
NC State University
Raleigh, NC
najmeri@ncsu.edu

Munindar P. Singh
NC State University
Raleigh, NC
mpsingh@ncsu.edu

ABSTRACT

Cybersecurity breaches cause enormous harm to the safety, privacy, and prosperity of individuals and organizations. Many security breaches occur due to people not following security regulations, chief among them regulations pertaining to what might be termed *hygiene*—including applying software patches to operating systems, updating software applications, and maintaining strong passwords. Ensuring user compliance with such cybersecurity regulations has proved challenging at least in part due to the fact that actions to improve cybersecurity often interfere at odds with a user's work.

We apply multiagent systems concepts to model interactions between users and system administrators. Specifically, we capture the expectations regarding cybersecurity that users have of each other and system administrators have of users as norms. The normative framework brings forth the possibility of sanctioning through which norms for hygiene can be created and maintained despite a competing motivation for users to complete their work.

This paper investigates sanctioning mechanisms with respect to their success in establishing norms for cybersecurity hygiene. We motivate three main varieties of sanctioning mechanisms, specifically, group, individual, and peer sanctions. We empirically investigate the effects of these sanctioning mechanisms in promoting compliance with cybersecurity regulations as well as the detrimental effect of sanctions on the ability of users to complete their work. We do so by developing a game that emulates the decision making of workers in a research lab.

We find that individual sanctions are more effective in motivating people to be compliant with cybersecurity regulations that group sanction. Group sanctions are more detrimental on the ability of users to complete their work. Our findings have implications for workforce training in cybersecurity.

KEYWORDS

Normative systems; Norms; Sanctions; Agent societies

1 INTRODUCTION

As computing has spread into every part of our economies and personal lives, two related trends are apparent: (1) cybersecurity threats can place more and more of our welfare at risk; and (2) attackers have more to gain from successful attackers and therefore the number and variety of attacks is continually proliferating. At the same time, user behavior is a major problem behind successful attacks in that users inadvertently allow avenues through which attackers can penetrate into computer systems.

A particularly insidious form of attack is the Advanced Persistent Threat (APT) [6], in which an attacker can penetrate a computer system and use it to explore and identify valuable resources, and exploit them from within the system by exfiltrating information or violating the integrity of information or even encrypting information to prevent its use until a ransom is paid.

Recognizing the challenges in user behavior, system administrators in various organizations have responded by creating cybersecurity regulations, such as keeping operating systems and applications up to date, disabling unused network ports and services, and not sharing passwords. We term such requirements as *cybersecurity hygiene* [17]. However, compliance with regulations is rarely adequate and many security breaches continue to occur due to human factors such as people not following security regulations. As an illustration of the problem, Kafalı et al. [13] study documented healthcare security and privacy breaches (each of which affected 500 or more patients) with respect to user behavior in light of security and privacy regulations.

Singh [22] points out the importance of cybersecurity as an application domain for multiagent systems (MAS), especially in light of modeling human behavior. Normative MAS provide an appropriate way of thinking for APT-like cybersecurity attacks because the actions or nonactions (e.g., carelessness) of one user can affect the outcomes for other users. In essence, we would like cybersecurity hygiene norms to be established and maintained. Sanctions [18] provide a recognized means to promote establishment of norms but have not been studied in connection with cybersecurity.

Contributions. To this end, we investigate how sanctions can promote cybersecurity hygiene. Specifically, we investigate two research questions in reference to three types of sanctions group, individual, and peer, which we explain below.

- How effectively does a sanction type lead to improved cybersecurity hygiene?
- How detrimental is a sanction type to user productivity?

Approach and findings in brief. We develop a game to simulate real-life work setting, such as a corporate office in which where workers complete assigned tasks while using computers. Each player assumes the role of an office worker. Each player is challenged to complete assigned tasks (captured as points earned) along

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 2019, Montreal, Canada

https://doi.org/doi

with maintaining the security of their computer. Failure to complete the security tasks may attract sanctions, causing loss in points earned or loss in opportunity to earn points.

We conduct several experiments in this setting. We find that workers complete more tasks and are sanctioned less in individual sanction setting as compared to group sanction setting.

Organization. The rest of this paper is organized as follows. In Section 2 we discuss an example elaborating on individual, group, and peer sanction. Section 3 presents a summary of related literture in normative multiagent system and game theory. We also discuss a multiagent simulation framework which analyses the effect of sanctions. Section 4 introduces the conceptual model which will be used to investigate how sanctioning affects security related behavior. Section 5 discusses the web-based game developed based on the conceptual model introduced in Section 4. Section 6 presents the design of the experiments conducted on MTurk. Section 7 presents the results of our experiments in relation to our research hypotheses. Section 8 finishes with conclusions derived from this study, threats to validity, and discuss future research directions.

2 EXAMPLES: INDIVIDUAL, GROUP, PEER SANCTIONS

Individual and group sanctions are applied by an administrator, a designated party who has the responsibility of monitoring and sanctioning, and peer sanctions are applied by users.

Example 2.1 (Zumbl Inc). Consider Alex, Bob, Charlie, and Dave, who are software developers working at Zumbl Inc. Each of them has a workstation connected to the same network. Software developers are tasked with completing projects. They use various tools available on their workstation to complete the project they are assigned. Each software developer has a different risk attitude and other personality traits. Additionally, Zumbl Inc. has defined cybersecurity regulations such as updating passwords periodically, installing security patches, and so on. The developers are expected to comply with these regulations. Erin is an IT administrator who looks after compliance with cybersecurity regulations.

Consider a case where Alex does not patch his workstation in time against an OS vulnerability. Erin observes that Alex's workstation was not patched. She disconnects it from the local network so that other workstations are not at risk, and patches it. Erin takes some time to patch the computer, during which Alex cannot work. Erin disconnecting Alex's workstation is an example of an *individual sanction* where the person who failed to comply with the regulation is sanctioned [18].

Alternatively, on noticing that Alex's workstation was not patched in time and could have affected other workstations on the same network, Erin along with disconnecting Alex's workstation, disconnects Bob's, Charlie's, and Dave's workstations to look for security breaches. Erin disconnecting all the workstations prevents all software developers from working on their respective tasks. This is an example of a *group sanction* [24]. Here, instead of disconnecting only Alex's workstation, who did not patch his computer on time, Erin disconnected all the workstations on the network.

Consider another case where Alex has not patched his workstation. Bob notices that Alex has not patched his workstation and fears that an exploit of the vulnerability on Alex's workstation could result in all the workstations getting compromised. Bob frowns at Alex and says that Alex patch his workstation, suggesting that all the workstations on the network would be at risk. Bob's frowning is an example of a *peer sanction* [12, 18].

3 RELATED WORK

We describe selected research from the following areas of relevance to this work.

Normative MAS. Mahmoud et al. [15] argue that negative sanctioning are a means for enforcing social norms with the assumption that agents applying such sanctions have unlimited resources—clearly false in the real world. Mahmoud et al. propose a resource-aware adaptive punishment technique, which they evaluate it in simulation. They showed their proposed punishment technique enables norm establishment with larger neighborhood sizes than static and the original adaptive punishment mechanisms. Sanctions are also used to promote norms in social contexts with privacy-aware agents [1].

We adopt the idea from approaches for learning norms [2, 4, 7, 11, 12, 14, 16, 19] to treat sanctions as negative rewards.

Cybersecurity. Du et al. [10] describe a multiagent simulation framework with agents performing security and research tasks in a lab under the scrutiny of a manager. They found that for individual and group sanctions, a lower observability of violations leads to lower compliance and higher task completion. Although we have similar motivations to theirs, we conducted empirical studies with humans based on a game we implemented. Our game supports peer sanctions, which their work lacks.

Game design. Several applications incorporate rewards via reputation points, badges, and leader boards. We adopted gamification to obtain plausible results from users in a simplified scenario and to use the game as a workforce training vehicle. Deterding et al. [8] explain that using game design elements can motivate users and increase their activity and retention. A game enables learners to experience situations that are infeasible in the real world for reasons of safety, cost, and time [23]. Role-playing is an established technique for educational activities [9], and can foster intrinsic motivation [8].

4 THE GAME MODEL

This section delineates formally the conceptual model which investigates how sanctioning affects the security related behavior of humans. To further investigate our research questions we design a game based on the conceptual model. We discuss the elements of the game and how they are related to the conceptual model.

We conducted a study on Amazon Mechanical Turk (MTurk) [3] where we asked participants to play the game. We discuss the design, different phases, and results of the study.

4.1 Conceptual Model

To investigate the effect of the sanctioning to humans' security-related behavior, we model a multiagent system comprising agents who play different roles.

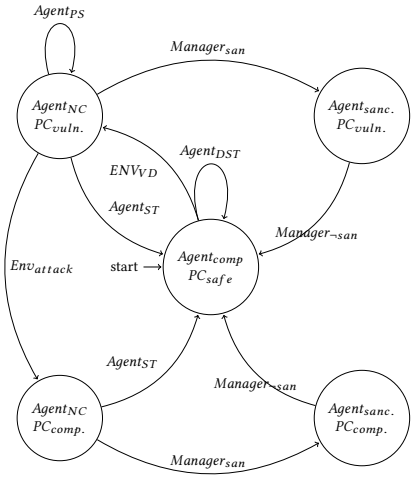

Figure 1: Conceptual model.

A *Worker* perform its domain-related tasks, i.e., tasks for which they have an external incentive to complete, such as office work than an employee is paid to. Each worker owns a PC, which is required for completing the domain-specific task. Meanwhile, a worker needs to perform security-related tasks such as patching the operating system, upgrading the firewall or changing the password, and so on, to keep the PC from being security compliant. In a system without sanctions, workers may lack the motivation of maintaining security and thus work on domain-specific tasks ignoring the security tasks. Whereas with sanctions integrated, the system has an administrator who monitors the system's security and takes measures to maintain security by imposing sanctions.

Outside of the system, there are potential *attackers*. They attack the system with the goal of compromising the PCs in the system. The degree of sabotage by the cyberattacks to the system depends on how secure the PCs are. Below is the formal definition of multiagent model with cybersecurity concerns.

Definition 4.1 (Secure Multiagent Model). The secure multiagent system model *O = {A, C, M, E}* contains four components: *A* is a set of workers who perform domain-specific tasks, security-related tasks and can sanction other workers. *C* represents a PC that is owned by the worker *A*. *M* is the manager in charge of the system, and environment *E* is everything besides the previous three entities, including the attackers.

4.2 Norms

A norm characterizes sound or *normal* interactions among the participants of a social group, reflecting their mutual expectations [21]. For instance, in the cybersecurity case, Zumbl Inc. is an organization, and the workstations are connected to form a network that seeks security. The IT administrator, Erin, monitors the network and take measures to ensure network security, by implementing a set of rules, such as patching and updating passwords. This set of rules define the security norms, which should be followed by the software developers in the company, reflecting the mutual expectation of cybersecurity.

Definition 4.2 (Norm). A norm is the directed normative relation between agents reflecting their mutual expectations from the system [21]. We define norms using a set of interaction rules:
N = {Interaction $\text{Rules}_{\text{agent}}$}

In a cybersecurity environment, norms could be *{Patching, Updating Passwords}*. Failure to follow the norms is a norm violation. For example, suppose Zumbl Inc. asks employees to update passwords every three months. An employee who updated his password more than 3 months ago is a violator. The employee may be sanctioned. We understand norms only from the perspective of the workers. We ignore the compliance or noncompliance for others.

4.3 Sanctions

An agent can learn about the norms by experiencing sanctions or observing sanctions being applied on others [20]. Sanctions can be positive or negative [18]. In real life scenarios, completing security tasks such as changing a password, updating an antivirus, and so on, does not have explicit rewards but not completing the security tasks can lead to negative consequences in the form of cyberattacks. In a similar manner, we do not consider explicit positive sanctions but employ negative sanctions as a consequence of not following the norm.

Definition 4.3 (Sanctions). The possible sanctions are:
S = {$S_{\text{Individual}}$, S_{Group}, S_{Peer}}

We consider peer sanction as a negative sanction from the agent in question to a peer because the former believes that the latter has failed to comply with the security norms. Group sanctions by manager can also motivate a compliant agent to peer sanction a noncompliant agent, to avoid a further group sanction. The sanctioner who issues a sanction has to be vigilant toward the security health of the PC of their peers and spend time to sanction their peers.

The manager can observe the security of all PCs and has the authority to sanction any worker in the system. The manager observes the norm violations based on its own observability. Depending on the specifics of the sanction, sanctionees have to face the consequences of the sanction. For instance, the sanctionee might be forced to spend time on fixing the security issue or the sanctionee PC may be evicted for certain time duration to fix the security issue.

4.4 Actions

Each agent has a belief set, tasks, and may perform different actions in different scenarios. The new state achieved after the action can be compliant or noncompliant, the latter potentially resulting in a sanction on the agent.

An agent at any time step can perform a domain-specific task, security task, observe and peer sanction other agents or do nothing because the agent was sanctioned.

Definition 4.4 (Agent's Actions). An agent's actions could be: Action_{agent} = {DST, ST, PS, NPT} where *DST* represents domain specific tasks, *ST* represents security tasks, *PS* represents productivity tasks, and *NPT* represents non productive tasks.

The manager observes (*Observe*) the state of the system and performs sanctions if the state is not norm compliant. We assume that the manager has limited observability and cannot observe the

security states of all the PCs at all times. This assumption is in line with real world scenario where the IT manager may not be able to observe all the PCs at all times in the organization.

Higher observability may quell the amount of repeated norm violations and increase the number of sanctions as the manager would be able to observe more violations, while low observability may allow agents to continually ignore normative expectations.

Sanctions could be imposed on an agent (*+Sanction*), or lifted from an agent (*-Sanction*). Lift of a sanction means the sanction no longer has an effect on the agent. For example, an agent is sanctioned because of security norm violations and it cannot use its PC for domain specific tasks. After the agent fixes its PC's security issues, the sanction is lifted and the agent could use its PC again.

Definition 4.5 (Manager's Actions). The manager's actions could be: $\text{Action}_{manager}$ = {Observe, +Sanction, −Sanction}, where *Sanction* could be *Individual Sanction* or *Group Sanction*.

Definition 4.6 (External Environment Actions). The external Environments actions could be: Action_{EE} = {Attack, Vulnerability Discovered}.

At any point, there can be a cyberattack attempting to compromise the PCs or there could be a new vulnerability discovered in one of the resources being used by PC.

4.5 States

Along with sanctions, PCs could have different states indicating its availability for performing tasks, as shown in Figure 2.

Definition 4.7 (PCs States). The set of PCs is represented as PC = $\{PC_1, \ldots, PC_n\}$. Each PC is associated with an agent. A PC's state model *CU = {Safe, Vulnerable, Compromised}* indicates whether it is safe, vulnerable, or compromised.

In the beginning, PCs are in the safe state. The PC stays in the safe state if the agent stays in compliant state. PC moves to a vulnerable state if the agent moves to a noncompliant state. An agent moves to a noncompliant state when a vulnerability is discovered and the agent has not yet fixed it. At this stage, the agent can be sanctioned for noncompliance or the vulnerability of the PC can be exploited by external agents and PC moves to a compromised state. The agent loses all the work on the PC. Agents are forced to perform security task at this stage to restore the PC to safe state again.

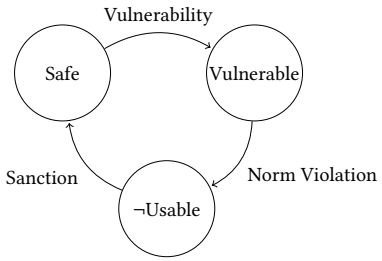

Figure 2: PC's usability model.

Definition 4.8 (Agent's States). The set of agents is represented as $AG = \{AG_1, \ldots, AG_n\}$. Each agent has an associated PC. An agent's state is one of these: $AU = \{Compliant, \neg Compliant, Sanctioned\}$.

Initially, an agent is in a compliant state. Later, agents perform actions. Based on the choice of actions, an agent moves to a noncompliant state or stays in a compliant state. If the agent moves to a noncompliant state, it can be sanctioned. Based on the nature of sanction, the agent needs to take certain actions or abide by certain limitations after which it moves back to a compliant state.

The interaction between states and action is shown in Figure 1. Circles shows the state of the agent and the PC. The caption on the arrow from one circle to another shows the action.

5 GAME DESIGN

To investigate how our model fits into real life situation, we designed a web-based game as shown in Figure 3, following the model.

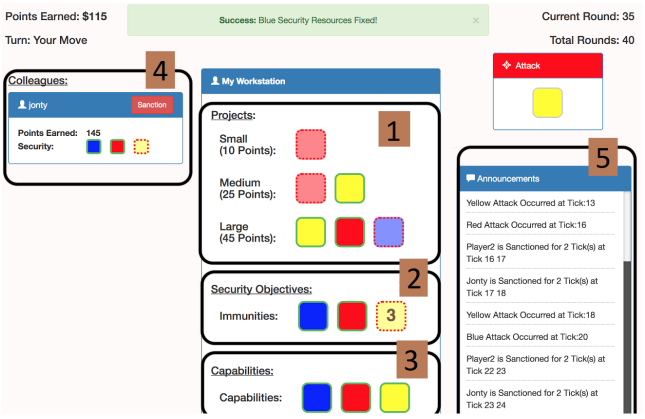

Figure 3: Game screenshot showing its five parts.

The game considers three resources: R_{blue}, R_{red} and R_{yellow}. The resources are represented by blue, red, and yellow color. Part 1 of Figure 3 shows the project section. The projects are equivalent to the domain-specific tasks. There are three types of project namely small, medium, and large with one, two, and three tasks respectively. Each task (T_i) is mapped to a resource (R_i) and requires that R_i be available for T_i to be completed. The availability of R_i is visible in Part 3 of Figure 3. The availability of resource is referred to as capability C_i in the game. The tasks that makes up a project is chosen randomly. For instance, in Figure 3, the medium project is made up by red and yellow resources. The choice of red and yellow resources are random. After a player completes all the tasks of a project, new resources are assigned to the project randomly.

Further there is an external environment which tries to compromise the resource via attacks. Each attack is directed toward a particular resource. After an attack (A_i), the player loses the immunity (I_i). For instance, yellow attack takes away yellow immunity.

The immunity is gained back by the player by completing corresponding immunity task, shown in Part 2 of Figure 3, for which there is a D_i. If the player does not complete the immunity task by deadline, the player can be sanctioned by the manager.

In case of an attack A_i directed at the resource R_i whose I_i is already lost, the player loses the C_i. The availability C_i can be gained back by completing the immunity task I_i. For instance, in case a yellow attack occurs and the player had already lost yellow

immunity, the player loses the yellow capability as well. After losing the yellow capability the player would not be able to complete yellow tasks in the projects until the player completes the yellow immunity task. A completed immunity task persists until there is another attack directed at the resource.

Every player in the game aims to earn the most number of points. The points in the game is equivalent to compensation that an employee of the organization gets for completing their daily tasks. Every player can see the current score of other players in the 'colleagues' section on the left side in Figure 3. The visible score of other players may motivate the players to perform better and improve their score. At the same time it can have negative implication. A player can continually peer sanction other players to prevent them from completing project task and gaining more points. To avoid the misuse of score information, the score can be completely hidden but that would have a negative implication in the form that players would not know how they are performing in comparison with other players and might lose the motivation to perform better.

The game is divided into 40 rounds, and in each round every player has to make a move before the game moves to the next round. Further, we explore each component of the conceptual model and see how they are mapped to the game.

Table 1: Game Parameters.

Parameter	Value
Attack probability	0.35
Manager observability	0.33
Small project score	10
Medium project score	25
Large project score	45
Tasks in small project	1
Tasks in medium project	2
Tasks in large project	3
Immunity task deadline	3

5.1 Norms

Table 1 shows that an immunity task has a deadline of three rounds i.e., the player gets three rounds to complete the immunity task after the immunity is lost without the risk of being sanctioned. Norm for a player is to complete the immunity task before the deadline ends while also completing project tasks to earn maximum points.

5.2 Actions

The game has different agents, each with different duties, and thus performing different actions. We discuss the actions of each agent.

5.2.1 Player Actions. In a round, a player can take one of the following actions in game:

Complete Project task. There are three types of projects based on the number of tasks required to complete the project. Table 1 shows the points awarded and number of tasks for each type of project. The player is awarded the score corresponding to a project only after he or she completes all the tasks of a project.

Complete immunity tasks. Players do not get awarded any points for completing immunity tasks but are required to complete it to avoid sanctions and to retain the availability of a resource.

Peer Sanction. Players could peer sanction other players if they observe that they have not completed their immunity tasks.

Do Nothing. When a player is sanctioned by the manager, the player can only pass the round by clicking on the pass button. Clicking on the pass button is an acknowledgment from the player that he has completed his turn.

5.2.2 Manager Actions. The manager observes the immunities of each player at the beginning of each round. If the manager observes that a player has not completed the immunity tasks past the deadline, the manager sanctions the player based on a probability. The sanction is applied at the beginning of the round. Table 1 shows that for one incomplete immunity task, the probability that the player will be sanctioned is 33 percent, and for two incomplete immunity task the probability increases to 66 percent. When a player has three incomplete immunity task, the probability increases to 100 percent. The sanctioning can be a group or individual sanction, depending on the means of sanction employed in the game.

5.2.3 Environment Actions. At the start of every round there is a probability of an attack, which is fixed throughout the game. We set the attack probability to 0.35 for all games, as listed in Table 1. It was chosen such that the number of attacks in a game are neither too high nor too low. High attack probability results in all players losing their capabilities thus forcing them to only complete immunity tasks. This would lead to same behavior by every player. Low attack probability might lead to a behavior where every player easily completes the immunity task instantly. We wanted the players to constantly choose between completing the immunity task and project tasks.

There are four types of attack in the game: A_{blue}, A_{red}, A_{yellow} and A_{black}. A_{blue}, A_{red} and A_{yellow} attack results in the loss of corresponding immunities I_{blue}, I_{red} and I_{yellow}. If the corresponding immunity is already lost, then the availability of the corresponding resource R_{blue}, R_{red} or R_{yellow} is lost. A_{black} attack is equivalent to the other three attack happening simultaneously. This presents players with cognitively most challenging goal to the player. The player has to complete all the immunity tasks within their respective deadlines.

To decide the type of attack, we choose a random number between 0 and 100. If it lies between 0 and 30 A_{blue} attack is chosen; for 30 to 60 A_{red} attack is chosen; for 60 to 90 A_{yellow} attack is chosen; and for 90 to 100 A_{black} attack is chosen. This distribution ensures that the probability of A_{blue}, A_{red} or A_{yellow} is equal whereas the probability of A_{black} is one third of other attacks.

The attacks in the game emulate the attack attempts made by the external environment. The attack frequency is the ability of external hacker on how frequent attack attempts the hacker make. Different types of attack corresponds to attacker trying to exploit different services used in the PC.

5.3 Sanction

The means of sanction in each game is configured to be either group sanction or individual sanction in addition to peer sanction. The

idea behind sanctioning is to have a negative reinforcement, to regulate the behavior of players.

There are two ways of sanctioning in the game. First is a group or individual sanction by the manager, where a player is forced to pass a certain number of rounds n; n is equal to twice the number of unfinished immunity task. The player passes the round by clicking on a pass button which appears after a player is sanctioned. The player cannot perform any other action for the number of rounds the player is sanctioned. After n rounds, the player gains back the immunity for which he or she was sanctioned.

Second is losing the availability of the resource. Each resource has a corresponding availability. Losing the availability means that player can no longer complete task associated with that resource until he or she completes the corresponding immunity task. For example if a player loses the red availability the player cannot complete red task anymore until the player completes the red immunity task. Losing the availability is equivalent to having the PC in a compromised state. Player also loses the completed task associated with that resource in the ongoing projects on losing the availability of the resource. A player loses the availability when an attack happens and the player does not have immunity for that attack. For example, a player will loose the red capability when a red attack happens and players does not have red immunity.

6 EXPERIMENTAL DESIGN

We conducted a study on MTurk where we asked participants to play our game. Thirty participants participated in the study playing 107 games. The group size for the games varied between 2–5. The study was approved by our university's Institutional Review Board. We collected an informed consent from each participant and provided a payment on completion the study. Each task that a participant performs in MTurk is called human intelligence task (HIT). The researcher approved the HIT if a participant participated in all the phases of the study irrespective of the participant's performance in the games played during the study.

We created a project task on MTurk. The participants could see instructions and an informed consent before they accepted our HIT. Participants need to sign the consent form to see a link to the game. We specifically instructed them not to participate in the study multiple times, since we posted several batches of HITs throughout the study. This was to mitigate the threat of learning. The game link redirected to our server where we hosted the game.

The study was conducted in slots of 60 minutes. We made multiple 60 minutes slots available for the study. Each participant after accepting the HIT selected a slot. The participant was expected to visit the game URL at the beginning of their slot and be available for next 60 minutes. To start with the study, participants were required to log in after visiting the game link. The username and password required for logging in was participants MTurk ID, which was communicated to them in the instructions of HIT. After logging in, participants were required to join a group chat whose link was available to the participants after logging in. The group chat made it easier for us to coordinate with the participants.

Survey. We asked participants to complete a survey which helped in assessing participants' personality and creative potential. First, we employ the Mini-IPIP (International Personality Item Pool) [5] scales to measure a participant's Big Five personality traits are Extraversion (E), Agreeableness (A), Conscientiousness (C), Neuroticism (N), and Openness to experience (O).

Second, we employ DOSPERT to assess participants' risk attitude [25]. It is a psychometric scale to assess risk taking in five content domains: financial decisions, health and safety, recreational, ethical, and social decisions. Both Mini-IPIP and DOSPERT are well-known scales. We choose these two scales because of their compactness. Longer alternatives increase the time spent on completing the survey and would leave us with less time for the gameplay.

Training. During training, each participant watched a five-minute video explaining the game. After the video, participants played two demo games, which were not included in the statistical evaluation. Demo games helped participants in getting acquainted. Participants were informed that these games will not be evaluated and were encouraged to familiarize themselves with different elements of the game. One demo game employed individual sanction and while other employed group sanction.

Gameplay. After training, each participant played four games. Two employed individual sanction and the other two employed group sanction in no particular order. Further, after each game, participants completed a short post-game survey to record their feedback on the sanction policy employed in the game.

Post Survey. After the gameplay, participants completed a post-study survey capturing their overall feedback.

We encouraged and rewarded players to give their best in the game by promising them a bonus based on the score in the game.

7 RESULTS AND DISCUSSION

We now describe the empirical evaluation that compared group and individual sanctioning mechanism based on the study conducted on MTurk. The idea of the study was to compare how people respond to each sanctioning mechanism and how their productivity in completing daily task is affected by these sanctioning mechanism.

Each participant played two games with group sanction and two games with individual sanction in span of 60 minutes. All the game parameters other than the sanctioning method were kept same throughout the study in all the games. These parameters can be seen in Table 1. We record every move made by a player in every game and evaluate this data to compare group and individual sanction with respect to each of the measures described above.

7.1 Metrics

To compare group and individual sanction we compute the following metrics. We test significance via the two-tailed paired t-test.

7.1.1 Compliance. It measures the frequency of an agent being compliant with a norm such as how frequently an employee of an organization is in completing the security tasks of changing the password or updating the anti-virus in a timely manner. In game, compliance means completing the immunity tasks before the deadline. We measure compliance via following measures:

Completed Security Tasks: In the game, after an attack, a player loses an immunity (S_i) and is given a deadline (D_i) to fix it. Fixing an immunity by deadline implies completion of security task.

Manager Sanctions: It counts the number of sanctions by the manager. A player is manager sanctioned if he or she does not complete an immunity task by its deadline. For individual sanction, it is calculated by the total number of sanctions issued to individual violators. For group sanction, it is calculated by the total number of sanctions issued to the group, irrespective of total number of players in the group. For example, for a game with three player group, an instance of group sanction is counted as one, not three.

7.1.2 Efficacy. It measures how productive participants are in completing their domain-specific tasks. Sanctions are detrimental to the productivity. We measure how detrimental group and individual sanctions are to the productivity. In the game, we measure efficacy via the following measures:

Score: Every time a player completes a small, medium, or large project, corresponding points are awarded. Score measures a player's productivity. Higher score indicates greater productivity.

Rounds Passed: Whenever a player is sanctioned by the manager or the peers, player is forced to pass a certain number of rounds. Passing a round means that player is not able to complete any productivity or security tasks in those rounds. This is equivalent to an IT administrator taking away the PC of an employee when the IT administration finds out that the employee has not applied security patches. Employee is not able to complete their daily task until the PC is returned by IT administrator.
Higher number of rounds passed mean players were non productive for those rounds and overall were less productive in completing the productivity tasks.

7.1.3 Resilience. It is measured by how quickly the system gets back to the state of being norm compliant. For example, when a vulnerability is discovered on a PC of an employee, how soon the vulnerability is fixed either by the employee or the IT administrator. In our game, We measure resilience via the following measure:

Rounds to gain back immunity: Number of rounds taken by a player to gain back an immunity after losing it to an attack.

7.2 Compliance

Figure 4 shows the box plot of the ratio of number of immunity task completed before deadline and number of immunity task available to each player in each game. Individual sanction offer slightly better (not significant) compliance than group sanction.

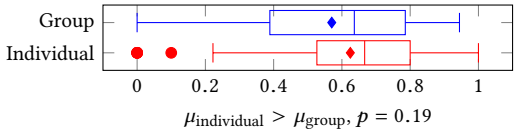

Figure 4: Ratio of immunity tasks completed before deadline to immunity tasks available.

We calculated the number of manager sanctions in each game and divided it by number of attacks in that game to normalize the number of sanctions. Figure 5 shows that on an average there were 23 sanctions for every 100 attacks in individual sanction games whereas there were 47 sanctions for every 100 attacks in group sanction games. This difference is statistically significant, while surprising, it is compatible with the result in Figure 4.

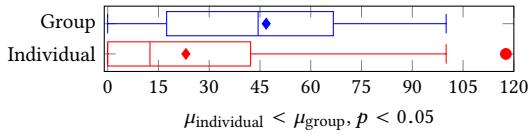

Figure 5: Number of manager sanctions per 100 attacks.

We found that there were 14 instances of peer sanction in games with group sanction and seven instances of peer sanction in games with individual sanction. This indicates that group sanction promotes peer sanction more than individual sanction.

After each game we asked the players, on a Likert scale of 1 (*not at all influential*) to 5 (*very influential*), how effective was the sanctioning mechanism. 77 percent of participants identified sanctions as a strong factor (4–5) in influencing their decisions.

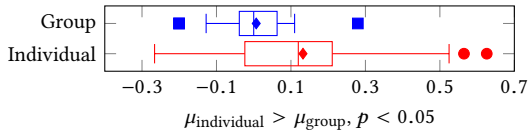

Figure 6: Difference between number of immunity tasks completed before deadline in first game and second game.

Each participant played two games each with individual and group sanction. We calculated the difference between the number of immunity tasks completed in the second game and the first game to measure the influence of sanction on the player. Figure 6 shows a box plot for the same. The difference has been normalized by dividing it with number of attacks in the game. On average a player completed 13 percent more immunity tasks per attack in the individual sanction games whereas in group sanction games number of immunity task completed in two games was almost same. This difference was found to be statistically significant ($p < 0.05$, signifying that individual sanction was more effective in motivating people to be security compliant as compared to group sanction.

To compare the compliance of participants based on their risk attitude, we divided the risk score of each domain obtained from DOSPERT survey into three categories: low, medium, and high. We calculated the number of manager sanctions and number of immunity tasks completed by players in each category. Figures 7 and 8 show that sanctions were more effective in promoting compliance among players with low and medium risk taking attitudes in social domain as compared to people with high risk taking attitude.

7.3 Efficacy

Figure 9 shows the average score of players in a game. Participants were more productive in individual sanction games. The average score in individual sanction games (372.5) was significantly greater than that obtained in group sanction games (333.85). The result is

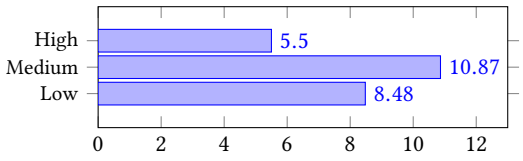

Figure 7: Immunity tasks completed per player grouped according to social risk taking personality.

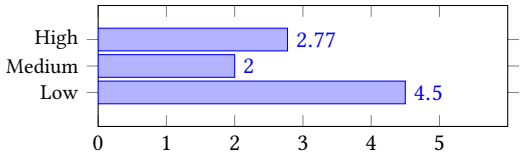

Figure 8: Manager sanction per player grouped according to social risk taking personality.

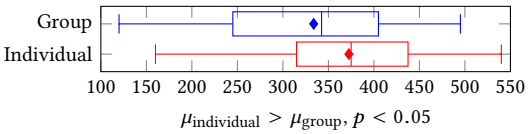

Figure 9: Average score.

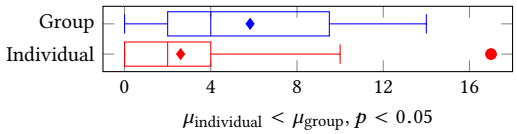

Figure 10: Average number of rounds passed.

not surprising as in group sanction games, compliant players were sanctioned because of non-compliance by others in the group.

Figure 10 shows the number of rounds a player passed in a game. On average, the number of rounds passed in group sanction is significantly more (double) than the of number of rounds passed in the individual sanction. This explains the low average score in Figure 9 for group sanction games compared to individual sanctions.

After each game we asked participants how detrimental the sanctioning mechanism was to completion of tasks in the game on a Likert scale of 1 to 5 with 1 being *very detrimental* and 5 being *not at all detrimental*. Figure 11 shows that participants found group sanction and individual sanction almost equally detrimental.

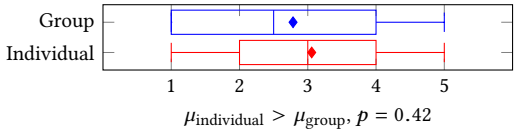

Figure 11: Perceived detrimental effect of sanction.

7.4 Resilience

We calculated the number of rounds a player took to gain back the immunity lost to an attack. The immunity could be gained back either by player completing the immunity task or being sanctioned by the manager which in turn fixes the immunities. Figure 12 shows while both group sanction and individual sanction motivated participants to complete the immunity task in the same round as both of them have mean close to zero, individual sanctions offer significantly better resilience ($p < 0.05$) than group sanction.

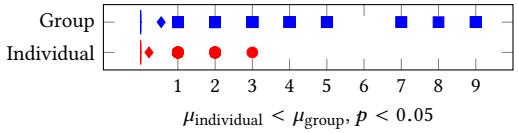

Figure 12: Rounds taken to gain back the immunity.

8 CONCLUSIONS

We investigate the effectiveness of group, individual, and peer sanctions in promoting cybersecurity hygiene and improving productivity. We conduct several experiments on the data collected through a cybersecurity game, played by workers on MTurk. We establish that that individual sanctions are more effective in enforcing compliance with the cybersecurity regulations than group sanction.

Workers are less productive in completing the projects tasks in group sanction games as compared to games with individual sanction but there are more peer sanctions in case of group sanction mechanism hinting toward a self-sustaining system.

8.1 Threats to Validity

Limited Sample Size. One limitation to the generalizability of our study is the sample size. Although we made 30 workers from MTurk participate in the study, they may not be representative of the larger population of the actual industry. Although large number of workers on MTurk signed up to participate in the study, the show rate at the designated time slot was low.

Motivation. Players may not be motivated to give their best and hence their game moves may not be well thought. For motivation, we promised a bonus based on their game performance.

Game Understanding. To make sure the game is understood well, we made the players watch a video explaining the game in detail. The first two games played by each player were not used in evaluation as these were to help players understand the game.

8.2 Future Directions

First, experiments with combination of group and peer sanction with variable manager observability is an interesting future direction. Observability of manager can be decreased gradually relying more on peer sanction to keep the agents security compliant. This type of system would scale better, but the effects of this on the productivity of participants is yet to be seen.

Second, group sanction along with peer sanction has the potential to function in a self sustaining system with no or little central agent involvement, where as in case of individual sanction there

will always be a requirement of central agent to make the system work in a security compliant manner.

Third, it will also be interesting to explore positive sanctions in future studies where an explicit positive reward is awarded for complying with the security regulations. The final score could be based on weighed average of number of security tasks and daily tasks instead of just daily tasks which is done in the current game.

REFERENCES

[1] Nirav Ajmeri, Hui Guo, Pradeep K. Murukannaiah, and Munindar P. Singh. 2017. Arnor: Modeling Social Intelligence via Norms to Engineer Privacy-Aware Personal Agents. In *Proceedings of the 16th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*. IFAAMAS, São Paulo, 230–238.

[2] Huib Aldewereld, Virginia Dignum, and Wamberto W. Vasconcelos. 2016. Group Norms for Multi-Agent Organisations. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 11, 2 (June 2016), 15:1–15:31.

[3] Amazon. 2011. Mechanical Turk. (2011). https://www.mturk.com/mturk/.

[4] Giulia Andrighetto, Jordi Brandts, Rosaria Conte, Jordi Sabater-Mir, Hector Solaz, and Daniel Villatoro. 2013. Punish and Voice: Punishment Enhances Cooperation when Combined with Norm-Signalling. *PLoS ONE* 8, 6 (06 2013), 1–8. https://doi.org/10.1371/journal.pone.0064941

[5] M Brent Donnellan, Frederick Oswald, Brendan Baird, and Richard E Lucas. 2006. The Mini-IPIP Scales: Tiny-yet-Effective Measures of the Big Five Factors of Personality. *Psychological Assessment* 18, 2 (2006), 192–203.

[6] Ping Chen, Lieven Desmet, and Christophe Huygens. 2014. A Study on Advanced Persistent Threats. In *Proceedings of the 15th IFIP TC 6/TC 11 International Conference on Communications and Multimedia Security (CMS) (Lecture Notes in Computer Science)*, Vol. 8735. Springer, Aveiro, Portugal, 63–72.

[7] Mehdi Dastani, Leendert van der Torre, and Neil Yorke-Smith. 2017. Commitments and interaction norms in organisations. *Autonomous Agents and Multi-Agent Systems (JAAMAS)* 31, 2 (01 March 2017), 207–249.

[8] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. 2011. From Game Design Elements to Gamefulness: Defining Gamification. In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments (MindTrek)*. ACM, Tampere, Finland, 9–15.

[9] Michele D. Dickey. 2007. Game design and learning: a conjectural analysis of how massively multiple online role-playing games (MMORPGs) foster intrinsic motivation. *Educational Technology Research and Development* 55, 3 (01 Jun 2007), 253–273. https://doi.org/10.1007/s11423-006-9004-7

[10] Hongying Du, Bennett Narron, Nirav Ajmeri, Emily Berglund, Jon Doyle, and Munindar P Singh. 2015. ENGMAS–Understanding Sanction under Variable Observability in a Secure Environment. In *Proceedings of Second International Workshop on Agents and Cybersecurity (ACySE)*. Istanbul, 15–22.

[11] Martijn Egas and Arno Riedl. 2008. The economics of altruistic punishment and the maintenance of cooperation. *Proceedings of the Royal Society of London B: Biological Sciences* 275, 1637 (2008), 871–878. https://doi.org/10.1098/rspb.2007.1558

[12] Chris Haynes, Michael Luck, Peter McBurney, Samhar Mahmoud, Tomáš Vítek, and Simon Miles. 2017. Engineering the Emergence of Norms: A Review. *The Knowledge Engineering Review* 32 (2017), e18.

[13] Özgür Kafalı, Jasmine Jones, Megan Petruso, Laurie Williams, and Munindar P. Singh. 2017. How Good is a Security Policy against Real Breaches? A HIPAA Case Study. In *Proceedings of the 39th International Conference on Software Engineering (ICSE)*. IEEE Computer Society, Buenos Aires, 530–540.

[14] Jiaqi Li, Felipe Meneguzzi, Moser Fagundes, and Brian Logan. 2016. *Reinforcement Learning of Normative Monitoring Intensities*. Springer International Publishing, Cham, 209–223.

[15] Samhar Mahmoud, Simon Miles, and Michael Luck. 2016. Cooperation Emergence Under Resource-Constrained Peer Punishment. In *Proceedings of the 15th International Conference on Autonomous Agents, Multiagent Systems (AAMAS)*. International Foundation for Autonomous Agents and Multiagent Systems, Singapore, 900–908.

[16] Mehdi Mashayekhi, Hongying Du, George F. List, and Munindar P. Singh. 2016. Silk: A Simulation Study of Regulating Open Normative Multiagent Systems. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*. IJCAI, New York, 373–379.

[17] Deirdre K. Mulligan and Fred B. Schneider. 2011. Doctrine for Cybersecurity. *Dædalus, the Journal of the American Academy of Arts & Sciences* 140, 4 (Fall 2011), 70–92.

[18] Luis G. Nardin, Tina Balke-Visser, Nirav Ajmeri, Anup K. Kalia, Jaime S. Sichman, and Munindar P. Singh. 2016. Classifying Sanctions and Designing a Conceptual Sanctioning Process for Socio-Technical Systems. *The Knowledge Engineering Review* 31, 2 (March 2016), 142–166.

[19] Charles Noussair and Steven Tucker. 2005. Combining monetary and social sanctions to promote cooperation. *Economic Inquiry* 43, 3 (2005), 649–660.

[20] Bastin Tony Roy Savarimuthu, Rexy Arulanandam, and Maryam Purvis. 2011. Aspects of Active Norm Learning and the Effect of Lying on Norm Emergence in Agent Societies. In *Proceedings of the 14th International Conference on Agents in Principle, Agents in Practice (PRIMA)*. Springer-Verlag, Wollongong, 36–50.

[21] Munindar P. Singh. 2013. Norms as a Basis for Governing Sociotechnical Systems. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5, 1 (Dec. 2013), 21:1–21:23.

[22] Munindar P. Singh. 2015. Cybersecurity as an Application Domain for Multiagent Systems. In *Proceedings of the 14th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*. IFAAMAS, Istanbul, 1207–1212. Blue Sky Ideas Track.

[23] Tarja Susi, Mikael Johannesson, and Per Backlund. 2007. *Serious Games : An Overview*. Technical Report HS- IKI -TR-07-001. University of SkÃűvde, School of Humanities and Informatics. 28 pages.

[24] Daniel Villatoro, Giulia Andrighetto, Jordi Brandts, Luis Gustavo Nardin, Jordi Sabater-Mir, and Rosaria Conte. 2014. The Norm-Signaling Effects of Group Punishment: Combining Agent-Based Simulation and Laboratory Experiments. *Social Science Computer Review* 32, 3 (2014), 334–353.

[25] Elke U Weber, Ann-Renee Blais, and Nancy E Betz. 2002. A Domain-Specific Risk-Attitude Scale: Measuring Risk Perceptions and Risk Behaviors. *Journal of behavioral decision making* 15, 4 (2002), 263–290.