ENGMAS – Understanding Sanction under Variable Observability in a Secure Environment

Hongying Du
NC State University
Raleigh, NC, United States
hdu2@ncsu.edu

Bennett Narron
NC State University
Raleigh, NC, United States
bynarron@ncsu.edu

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

Emily Berglund
NC State University
Raleigh, NC, United States
emily_berglund@ncsu.edu

Jon Doyle
NC State University
Raleigh, NC, United States
jon_doyle@ncsu.edu

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

ABSTRACT

Norms are a promising basis for governance in secure, collaborative environments—systems in which multiple principals interact. Yet, many aspects of norm-governance remain poorly understood, inhibiting adoption in real-life collaborative systems. This work focuses on the combined effects of sanction and observability of the sanctioner in a secure, collaborative environment. We introduce ENGMAS (Exploratory Norm-Governed MultiAgent Simulation), a multiagent simulation of students performing research within a university lab setting. ENGMAS enables us to explore the combined effects of sanction (group or individual) with the sanctioner's variable observability on system resilience and liveness. The simulation consists of agents maintaining "compliance" to enforce security norms while also remaining "motivated" as researchers. The results show with lower observability, agents tend not to comply with security policies and have to leave the organization eventually. Group sanction gives the agents more motive to comply with security policies and is a cost-effective approach comparing to individual sanction in terms of sanction costs.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*multiagent systems*

General Terms

Security, Human Factors

Keywords

Norms, Resilience, Liveness, Safety, Observability, Multiagent System

1. INTRODUCTION

Secure, collaborative environments often suffer from *ad hoc* implementations of policy developed by its governing principal through careful scrutiny of secure practices and reactionary measures to threat [5]. However, assessing the security of an environment is costly, and while having the clarity of hindsight, patching existing vulnerabilities does not ameliorate the damages caused by an attack. To address this issue, norm-based solutions for governance have emerged to introduce a more pliable means of developing policy [7, 14, 15]. While promising, attempts-to-date have failed to fully capture a low-level interpretation of a norm-governed system. Still, the pursuit of understanding the application of norms to system security has revealed a promising frontier for research.

The ultimate goal of our research is to draw nearer to a complete mathematical model of norms, so that we may understand their behavior as a form of governance in a system of heterogeneous, autonomous principals. More specifically, we are concerned with how these norms and policies may influence a system's liveness and resilience; however, we are still posed with the question: Can one predict the liveness and resilience of a system as a function of its norms and policies and the social environment? Further, do any trade-offs exist between liveness and resilience that may be influenced by norms and policies?

The answers to these questions are computationally hard and are, therefore, outside of our ability to address using a game-theoretic approach. As a result, we have designed and implemented an exploratory multiagent simulation, called ENGMAS, to address our questions. The simulation is a centralized system wherein autonomous agents perform tasks under the scrutiny of a governing principal. ENGMAS is regulated by multiple, adjustable settings, many of which may have profound effects on the outcome of the simulation; however, the primary focus of this paper is to explore the variable observability of the governing principal and the type of sanction type (individual or group) applied to norm violation.

We acknowledge two factors as possible influence on the resilience and liveness of a system: (1) a sanctioning agent's observability of its environment and (2) the means of sanction applied (group or individual).

2. TERMINOLOGY

The variability in the literature regarding the terms we use to describe our research warrants careful attention. Thus, we devote this section to defining our interpretations of each concept we intend to present.

2.1 Norms and Sanction

We recognize the term, *norms*, to describe "directed normative relationships between participants in the context of

an organization" [14]. For example, consider a university setting in which there exist multiple research labs nested within its various departments. Within each lab, there is a finite set of graduate students assisting with departmental research, utilizing security-critical artifacts (namely, PCs or machines) connected to a shared network. The network may be monitored by IT staff, whose role it is to ensure the safety of the system. In doing so, the network administrator may implement a directed set of agreements (including, but not limited to, patching the operating system, updating passwords, and maintaining a firewall) which must be met by each of the graduate student researchers in order to avoid some consequence. Alternatively, graduate students are also tasked with completing research in order to maintain grants, earn credit hours, or other obligations. Each student's advisor has an expectation of motivation directed towards the student.

Failure to comply with normative expectations is met with *sanction* (consequence for norm violation applied to a principal or group of principals) by a sanctioning agent (e.g., the IT staff in the previous example). A sanction may be positive or negative and manifested in reprimand or reward, respectively [11]; though, in the context of the university research lab environment, norm violations would result in negative sanction. When an individual principal is singled-out and censured for defecting against a norm, we recognize this as "individual" sanction. Alternatively, when sanction is applied to a group of individuals for the actions of some subset of that group, we recognize this as "group" (also known as "collective") sanction [8].

2.2 Safety and Liveness

Lamport is credited with the earliest acknowledgment of the safety and liveness compromise [9]. Following from his seminal work, we define *safety* as the stipulation that some action (or inaction) does not cause the system to enter a bad state. In our university lab example, we describe the safety concerns of the system administrator as his or her desire to maintain a secure system. The safety of the system is in direct correlation with the magnitude of security compliance by the graduate students in the lab. That is, if the students forego the norms that describe the security measures they are expected to perform, the system is likely to enter a compromised state. In response, the administrator may shut down some subset of network connections that he or she deems responsible for the threat, an action that would directly benefit the desires of the administrator but would be devastating to the system's liveness.

Liveness is characterized by maintaining a good state, or more specifically, a state of production [2]. Counter to the safety example, liveness can be inhibited by safety precautions and serves the interest of the research advisers. Both enacting safety measures and conducting research are costly endeavors, and limitation of resources (i.e., time) will almost certainly result in the investment of one obstructing the ability to perform the other. In the lab, we would witness students attempting to adhere to security policy at the expense of research load, or vice versa.

2.3 System Resilience

Multidisciplinary interpretations of *system resilience* (in Economics, Anthropology, Social-Ecology, and Resilience Engineering, among others) abstract to several meaningful definitions, each with respect to its own application (e.g., [1, 4, 6, 12, 13]). For the purpose of our research, we align our understanding of resilience with Sheridan, who describes it as the ability to "recover and restore the system to the original state or, if need be, some acceptable state that is different but still safe" [13]. We do not attempt to define the notion of an "acceptable" state and further recognize it to be an arbitrary factor. The acceptability of a state is subject to the opinion of any active principal in the system, and more often than not, these views are conflicting. If we consider the dichotomy presented earlier between the expectations of the system administrator and those of a graduate student adviser, there is no guarantee that there is any satisfiable union between any set of states that both parties would deem "acceptable". For this reason, our metric for resilience will be determined by a variable dependent on both safety and liveness, named *load*.

Load is defined by the ratio of the actual amount of work completed against the total potential, aggregated over the sum of the number of active, performing agents and the number of the non-active agents in the system. It is representative of the productivity of a system at a single time step. Tasks performed by agents which contribute to the safety of a system are not considered to be "productive", and thus, are not a component in the calculation of load. However, such tasks do effectively assist in maintaining a safe state and allow productive actions to occur. In our lab analogy, performing research would qualify as a productive task, while implementing security features would directly promote the safety of the environment.

Given load, we calculate the resilience of a system by how quickly it recovers from successive disturbances (more specifically, norm violations). This phenomenon is best illustrated by the waxing and waning of load over time. As active principals neglect to fulfill expectations regarding the safety of the system, the resulting downtime will likely quell the productivity of the system. After agent behavior is modified via sanction, a resilient system is likely to efficiently recover to some arbitrarily acceptable state, while a rigid system most likely will not.

2.4 Observability and Efficiency

The primary focus of this paper is to understand system resilience as a result of the combined effects of the sanctioner's observability of his or her environment coupled with the type of sanction applied to principals responsible for norm violation. More explicitly, we are interested in the scenarios of the sanctioner applying individual sanction on identified defectors, or group sanction despite the identities of the defectors under various observabilities.

Each of the aforementioned cases has underlying implications associated with it. For example, the system administrator (i.e, the sanctioner) for our university lab is tasked with sanctioning an agent after he or she realizes that the network has been compromised. If the administrator seeks to identify the specific party that caused the vulnerability, the influence of the individual sanction may not have much affect on the rest of the group. Alternatively, group sanction may lead to vigilance among agents, which could promote whistle-blowing. Further, the more transparent the lab is to surveillance, the more efficiently sanctions may be applied; this may quell the amount of repeated norm violations by a single agent, while partial observability may allow agents to

continually ignore normative expectations. We acknowledge these factors as essential to understanding system resilience and have designed a simulation to apply these conditions to a university lab setting.

3. THE SCENARIO

To further investigate our proposed research, we extend our lab scenario to a multiagent simulation, called ENGMAS. A more formal description of our scenario involves a university graduate research lab (congruent to our running example) and its constituent student researchers, represented by agents. The system contains three types of entities:

(1) A lab, or an organization, wherein student agents perform activities.

(2) A set of Student Agents, each representing a student researcher who controls a PC in the lab. Each agent must respond to the tasks assigned to it using its PC, and will be sanctioned if it violates the norms of the system.

(3) A special, centralized agent named Carlos, who is responsible for applying sanctions to the agents in the lab.

3.1 Carlos

Agents have different duties and, therefore, play different roles within the system. Carlos is in charge of sanctions, thus his responsibilities are to:

- Observe, or monitor the lab's network for any visible norm violations. *Observability (O)* varies on a scale, ranging from 0% (inability to observe norm violations) to 100% (ability to observe all norm violations). For example, if Carlos has observability of 80%, then at each time step there is an 80% chance he will discover a norm violation, independently, given a set of student agents who have failed to comply with security norms.

- Perform sanctions as soon as a norm violation is caught. We represent sanctions as shutting off network access of PCs in the lab for a certain amount of time, which prevents threats to PCs from spreading. After sanction, each student agent whose PC health is below a threshold has to fix its PC. Carlos could perform one of the following two kinds of sanctions:

  - *Individual Sanction*, the PCs controlled by agents who violate norms are disconnected from network for a constant period of time.
  - *Group Sanction*, all PCs in the lab are disconnected from network for a constant period of time, despite the identity of defecting agent.

  There are expenses for Carlos to discover norm violations, which are different for the two kinds of sanctions. For individual sanction, Carlos needs to find out who are the defectors, which costs more than simple observation of norm violations without figuring out the defectors in group sanction.

3.2 Student Agents

Student agents could perform the following actions:

- Domain-related actions, which includes research related actions to fulfill their research responsibility, and security related actions to ensure the security of their PCs and thus to protect the integrity of the lab environment. Security related actions include: patching the operating system, turning the firewall on and off, updating passwords, installing, turning on or off, and updating the anti-virus software.

- Observe, or monitor other student agents' actions. Agents in the lab have the ability to observe the actions of others, though not particularly well. For example, under the constraints of low observability, Carlos may not be capable of observing all activity on the network; thus he may rely on student agents to report norm violations to him. In this case, student agents must be able to observe other agents' or their neighbors' (i.e., agents within close proximity) actions.

- Communication. To report a norm violation to Carlos in the above case, student agents have a channel for communication with Carlos. They may send messages reporting other students for norm violation.

There are two types of tasks (actions that agents are obligated to perform) that may be assigned to student agents:

- A research task represents the research work that a student must finish (for example, writing a paper, reading papers, etc) in order to fulfill the expectations of his or her adviser.

- A security task represents the security precautions a student must take to ensure that his or her PC is safe. It could be one of the security-related actions or any combination of them.

Each task type has two attributes:

- Duration, or the amount of time it takes for a student agent to complete a task. Compared to a research task, a security task is much less time-consuming for an agent, as represented in reality (e.g., doing homework takes much longer than changing a password).

- Deadline, or the amount of time a student agent is allowed to complete a particular task. If it is unable to complete a task by the deadline, the agent's health (for failure of a research task) or PC health (for failure of a security task) will decrease. For each task, deadlines are allotted with ample time for a student to complete it. For research tasks, we determine the deadline by applying a fixed coefficient, c, to the task duration. For example, if a research task consists of five time steps, theoretically an agent is given $5 * c$ time steps to complete it. We consider a time step as the smallest time unit and round decimals to the nearest integer when calculating deadlines.

Each student agent has five attributes:

- *Agent Health* represents a student agent's health and is influenced by the completion status of research tasks assigned to the agent. Agent health is subject to change during the simulation for the following reasons: (1) If an agent finishes a research task before the deadline, its agent health is increased accordingly. (2) If an agent is unable to finish a research task before the deadline, its agent health is reduced accordingly. If an agent's

health reaches a fixed low value, it must leave the organization or lab. If an agent leaves, it is no longer able to perform any action and will never return to the simulation and we consider it as non-active.

- *PC Health* represents the PC's health and is influenced by the completion status of security tasks assigned to the agent: (1) If an agent finishes a security task before its deadline, its PC health is increased accordingly. (2) If an agent is unable to finish a security task before its deadline, the agent has committed a norm violation, and its PC health is reduced accordingly. If the PC health drops under a fixed value, the PC has been compromised and is unusable until it is sanctioned by Carlos and starts to fix his PC.

- *Preference* is the probability that an agent will choose to begin working on a research task first, under the condition that it has both a research task and a security task to perform. For example, if an agent's preference is 80% and it has both a research task and a security task on its task list, it will have an 80% chance of considering the research task and 20% chance of considering the security task. Note that by "considering", the agent does not start working on the task. It merely implies that the agent has decided which task to attempt. The probability that an agent begins working on a task is dependent upon one of the following two attributes.

- *Research Motivation* is the probability that an agent will choose to start a research task, given that the agent with both tasks has considered it, or the agent only has a research task. After an agent fails to finish a research task, its research motivation increases, indicating that it is motivated to start a research task early next time.

- *Security Compliance* is the probability that an agent will choose to start a security task, given that the agent with both tasks has considered it, or the agent only has a security task.

Preference, research motivation, and security compliance together dictate which tasks are being started, if any, during a given time step. The initial values of preference, research motivation and security compliance for each student agent are generated via normal distributions.

4. SIMULATION AND EVALUATION

4.1 Assumptions and Settings

Due to the complexity of real-world scenarios, we have made several assumptions in our experiment. We do not trivialize the significance of these variables, nor do we recognize them as arbitrary. In future work, we intend to determine valid replacements through further research, experimentation, and data collection. However, as the nature of our simulation is exploratory, we have intuitively assigned values to these variables, which are held constant throughout all treatments and runs of the simulation. Further, we will use the term, "tick", to denote a single time step or the smallest time unit defined by users. One tick could be of different time length in different applications.

Initial values of agent health and PC health are both 100. Research tasks are assigned every three ticks; security tasks are assigned every seven ticks.

Research tasks are assigned only to active agents (i.e., agents whose agent health are greater than zero), regardless of whether their PCs are down (PC health is zero) or not. For any research task, we generate a number from a normal distribution as its duration. Its deadline is decided by the coefficient and duration as explained before. If an agent's PC is down, it will not be able to perform tasks and its PC health cannot decrease any further. If an agent leaves the lab, its research tasks are cleared, i.e., no more research tasks are on its list.

Security tasks are assigned to active agents with PCs not down. To simplify the simulation, we treat security-related actions all as abstract security tasks and do not distinguish between different security-related actions. For a security task, since it takes relatively less time to finish, we assume each security task actually takes only one tick. We allot the agents a static seven ticks for a security task, since it only takes one tick to complete a security task and assignment is less frequent than research tasks.

For simplicity, we treat all the student agents in the lab as one group, and ignore the implementation of their observation and communication actions. Appendix shows a summary of the parameter values not presented here.

4.2 Runtime Actions

At each tick, the following actions occur:

- New tasks are assigned to the agents in the lab, if possible. We assume a research task may only be assigned to agents who do not already have assigned research tasks with deadlines exceeding the current tick. For example, an agent has a research task with deadline equal to tick 9, then no research task will be assigned to it until tick 10. In this way, we eliminate the possibilities that agents are assigned too many tasks beyond their capabilities. Half of all agents or all available agents, whichever are lower in number, are assigned research tasks. Every agent is eligible to be assigned a security task. An agent may work on at most one task type per tick.

- Student agents attempt to perform tasks. There are two possible status at each tick: the agent is working on an incomplete task or its deadline is not yet expired, so he continues working on the task; or the agent is not currently working on a task, so it must choose a task to perform. In the latter case, there are four extended possibilities:

  (1) If the agent has no task on its list, then it will rest.

  (2) If the agent has only a research task, it chooses whether to do the assigned research task or not, decided by its research motivation.

  (3) If the agent has only a security task, it chooses to do the assigned security task or not, decided by its security compliance.

  (4) If the agent has both the research task and the security task, it makes the decision following this procedure: the agent chooses to consider which task, according to the preference attribute. We call this chosen task the "preferred task". The agent will then choose

whether to start working on the preferred task according to research motivation or security compliance, respectively. If the agent chooses not to start the preferred task, then the agent rests.

If an agent chooses to begin working on a task, we assume the agent will continue to work on it until it is completed, unless the agent leaves the lab, it's being sanctioned, the agent's PC is down, or trying to fix its PC after being sanctioned.

- PC health decays daily proportional to the ratio of the number of agents whose PC health is below 80 over the total number of PCs in the lab. Maximum decrease for a single day is limited to a value of five. This mimics the viral influence of PCs in a bad state on other PCs in the lab, as a network with PCs that have security issues may be more vulnerable and thus influence other PC's health in the network, as well.

- Carlos may or may not observe norm violations, based on observability. As long as he discovers norm violations, he issues a particular sanction depending on his sanction type. We assume a sanction takes one tick. After Carlos issues a sanction, the sanctioned agents' PCs are shut off network access for one tick. After sanction, each agent's security compliance is increased by a fixed percentage (25% in our case), and the agents are forced to begin restoring their PCs to an acceptable state. PC restoration occurs incrementally following sanction, with each agent's PC health increasing by 15% at each time tick until it reaches 80, at which point the agent may resume completing tasks.

- The research motivation and security compliance attributes of each agent decreases under certain conditions in order to mimic the behavior of agent complacency over time. In our simulation, if an agent is not working on a research task and its health is above 60, its research motivation decreases by 0.01 at that tick. Also, if an agent has not been sanctioned for 45 ticks, its security compliance decreases by 0.01. For example, if tick 1 is the last time an agent gets sanctioned, its security compliance starts to decrease at the end of tick 47.

4.3 Metrics

The following metrics are measured over the course of the simulation:

- *Liveness*, or *System Research Motivation (M)*: Calculated as the median value of the averages of all the agents' research motivation values at each tick of the simulation.

- *System Security Compliance (C)*: Similar to system research motivation, calculated as the median value of the averages of all the agents' security compliance values at each tick of the simulation.

- *System Load*: Load is calculated as the ratio of the number of agents who are actively performing a research task at each time tick over the sum of the number of agents who have research tasks on their lists and the number of non-active agents. It measures the percentage of research load by all agents capable of production. We define "System Load" as the median load over the entire simulation period.

- *Resilience*: Resilience is measured by how quickly the system can recover after successive norm violations. In our experiment, it is measured by the average time it takes for the system to recover to an acceptable state after falling into a bad state. We define a system with load <= 0.4 as being in a bad state and recognize it as recovered if load increases to a value >= 0.7. The defined threshold is arbitrarily-assigned and is intended to allow the simulation to capture the slope of rebounding curves. We record all occurrences of this phenomenon and average them over the course of the entire simulation, regarding smaller values as more resilient than larger values.

- *Total Research Tasks*: The number of research tasks that are completed in addition to those that are not completed (and past deadline) by the end of the simulation.

- *Completed Research Tasks*: The number of completed research tasks by all the agents throughout the simulation.

- *Total Security Tasks*: The number of security tasks that are completed in addition to those that are not completed (and past deadline) by the end of the simulation.

- *Completed Security Tasks*: The number of completed security tasks by all the agents throughout the simulation.

- *Violations*: The total number of norm violations committed throughout the run of the simulation. Note that if an agent didn't complete a security task while it's fixing its PC after being sanctioned, it doesn't count towards a norm violation because the agent is doing security measures. Thus, it is possible that the "total security tasks" value may not be equivalent to the sum of completed security tasks and violations.

- *Sanctions*: The total number of sanctions issued during the simulation. For individual sanction (S_i), it is calculated by the total number of sanctions issued to individual violators. For group sanction (S_g), it is calculated by the total number of sanctions issued to the group, despite the number of agents in the group.

4.4 Evaluation

We averaged results of each metric over 50 simulations, each with 1000 ticks (or until all the agents leave the organization, whichever the earliest), under three agent population sizes (100, 500, and 1,000). Tables 1–6 illustrate the results of our treatments with metrics as defined above.

4.4.1 Network Sizes

We ran our simulation for three different network sizes: small (100 agents), medium (500 agents), and large (1,000 agents). Results demonstrate that for variable observability (from 0% to 100%), the size of the network has no quantifiable affect on system research motivation, system security compliance, or system load.

Table 1: Individual Sanction Results for 100 Agents over 50 Simulations
O – Observability, M – System research motivation, C – System security compliance,
S_i – Sanctions in individual sanction, S_g – Sanctions in group sanction

O	M	C	Load	Resilience	Research Tasks		Security Tasks			
					Total	Completed	Total	Completed	Violations	S_i
0%*	0.68	0	0	1.73	3216	1330	2087	1137	950	0
20%*	0.63	0.28	0.43	1.73	3804	1767	2814	1829	974	194
40%	0.44	0.45	0.71	2.42	11487	8524	14243	9656	4587	1833
60%	0.43	0.56	0.72	2.33	11489	8619	14268	10752	3515	2111
80%	0.43	0.64	0.72	2.65	11497	8675	14271	11365	2907	2331
100%	0.42	0.7	0.73	2.51	11498	8716	14274	11741	2533	2533

* – Simulations stop at ticks between 300 – 450, # – A small portion of simulations stop before 1000 ticks

Table 2: Group Sanction Results for 100 Agents over 50 Simulations

O	M	C	Load	Resilience	Research Tasks		Security Tasks			
					Total	Completed	Total	Completed	Violations	S_g
0%*	0.68	0	0	1.73	3207	1324	2074	1136	938	0
20%#	0.44	0.74	0.7	2.5	10510	7675	12603	10193	2400	26
40%	0.43	1	0.75	2.12	11501	8698	14282	12487	1795	56
60%	0.44	1	0.76	1.93	11490	8584	14281	12325	1956	86
80%	0.46	1	0.76	1.79	11493	8470	14279	12107	2172	115
100%	0.48	1	0.77	1.68	11504	8355	14277	11908	2369	142

Table 3: Individual Sanction Results for 500 Agents over 50 Simulations

O	M	C	Load	Resilience	Research Tasks		Security Tasks			
					Total	Completed	Total	Completed	Violations	S_i
0%*	0.71	0	0	1.52	16077	6651	10438	5677	4761	0
20%*	0.65	0.28	0.33	1.52	19009	8815	14021	9133	4835	969
40%	0.44	0.45	0.71	1.52	57454	42636	71227	48317	22911	9162
60%	0.43	0.56	0.72	1.52	57476	43123	71342	53773	17568	10545
80%	0.42	0.64	0.73	1.52	57471	43377	71362	56831	14531	11628
100%	0.42	0.7	0.73	1.55	57481	43552	71379	58746	12633	12633

Table 4: Group Sanction Results for 500 Agents over 50 Simulations

O	M	C	Load	Resilience	Research Tasks		Security Tasks			
					Total	Completed	Total	Completed	Violations	S_g
0%*	0.71	0	0	1.52	16068	6642	10423	5683	4740	0
20%#	0.44	0.73	0.71	2.27	52151	37814	62148	50020	12108	25
40%	0.43	0.99	0.76	1.95	57480	43474	71405	62416	8989	56
60%	0.44	1	0.76	1.79	57484	42964	71403	61673	9730	85
80%	0.46	1	0.76	1.68	57461	42374	71392	60618	10774	113
100%	0.47	1	0.77	1.65	57466	41747	71386	59502	11884	142

Table 5: Individual Sanction Results for 1000 Agents over 50 Simulations

O	M	C	Load	Resilience	Research Tasks		Security Tasks			
					Total	Completed	Total	Completed	Violations	S_i
0%*	0.72	0	0	1.51	32125	13269	20917	11346	9571	0
20%*	0.65	0.28	0.26	1.51	37862	17514	27871	18140	9626	1924
40%	0.44	0.45	0.71	1.51	114867	85240	142467	96592	45875	18325
60%	0.43	0.56	0.72	1.51	114977	86257	142682	107523	35159	21093
80%	0.42	0.64	0.73	1.51	114968	86761	142730	113631	29099	23277
100%	0.42	0.7	0.73	1.51	114986	87125	142755	117479	25276	25276

Table 6: Group Sanction Results for 1000 Agents over 50 Simulations

O	M	C	Load	Resilience	Research Tasks		Security Tasks			
					Total	Completed	Total	Completed	Violations	S_g
0%*	0.71	0	0	1.51	32136	13277	20892	11351	9541	0
20%#	0.44	0.74	0.73	2.27	106019	78166	128307	104718	23531	26
40%	0.43	1	0.76	1.88	115006	86943	142823	124917	17905	56
60%	0.44	1	0.76	1.73	114953	85927	142805	123414	19391	84
80%	0.46	1	0.76	1.66	114995	84784	142785	121245	21540	114
100%	0.47	1	0.76	1.65	114947	83501	142773	119023	23750	142

For individual sanction, resilience has less meaning in our setting. In a medium or large network, there are between 0 and 1 instances that the phenomenon occurs where system load increases from the lower threshold, <= 0.4, to the higher threshold, >= 0.7. The <= 0.4 value usually occurs at the first tick of the simulation. For individual sanction in a small network, load varies more than that in the other two networks, exhibiting instances where there are 0 – 9 instances where load increases from <= 0.4 to >= 0.7. For group sanction, the number of intervals is almost equal to the number of sanctions. Interestingly, resilience for group sanction slightly decreases with increase in network size from small to medium, and stabilizes as the network grows to large.

4.4.2 Observability

For both sanction types, at lower observability agents have a high system research motivation. However, at a higher observability (> 20% for Group Sanction and > 30% for Individual Sanction), there is no significant difference in system research motivation. Also, with increase in observability, the system security compliance increases under any sanction in a network of any size since larger observability leads to more sanctions.

At lower observability (< 20% for Group Sanction and < 30% for Individual Sanction), agents tend not to complete security tasks due to less sanctions. As a result, their respective PC health values decrease to zero (PC health threshold of compromising) eventually. Despite high research motivation, agents continually fail to complete research tasks due to zero PC health. Later, agent health values of all agents drop to zero (the agent health threshold for leaving the lab) because of unfinished research tasks, and the simulation ends before 1000 ticks, as indicated by * and # in the tables.

4.4.3 Sanction Types

Group sanction leads to slightly greater system research motivation than individual sanction when observability > 40%. For smaller observability, many simulations stopped before 1000 ticks, thus it is not meaningful to compare their values to those with larger observability. Further, student agents are more willing to comply with the security policies under group sanction, with violations occurring less frequently when the sanctioner has higher observability and more frequently with lower observability. It is also evident that individual sanction leads to 6–20 times greater total number of sanctions than group sanction, which suggests that the cost of individual sanction is much higher than that of group sanction. Thus group sanction may be adopted as a cost-effective approach.

4.4.4 Tasks

In group sanction, student agents are driven to give more priority to security tasks when sanctioned, thus leading to more completed security tasks than in individual sanction. During the period that the agents are trying to fix their PCs after being sanctioned, they are behind schedule on their research tasks, and as a result, less research tasks are completed.

5. RELATED WORKS

Our interest in the costs related to sanction arose as a result of surveying literature concerning low-level representations of normative relationships. Dissatisfied with our findings, we constructed an analogy (the research lab scenario) that encapsulated the aspects of norms we were interesting in investigating. While tuning the variables of our simulation, we began to question the proper mode of sanction to impose on the agents for norm violation, which lead us back to the literature for answers. With no applicable response, we were determined to formulate our own analysis.

Previous works in observability and cost-efficiency are primarily concerned with decentralized systems (i.e., no administrating principal), where agents participate in a cooperative game (see [3, 10, 16]). Mahmoud et al. impose a game on a set of agents who are required to make binary decisions to either comply or defect when presented with a norm. Norm violations are met with punishment, a form of consequence specifically targeted at destabilizing the economic gain of the defector [16]. The agent's decisions also affects its reputation and sets precedence for how it is to be treated later in the simulation by other agents. The ability for an agent to judge another based on reputation is dictated by that agent's ability to observe its environment. Mahmoud et al. conclude greater observability yields stronger norm compliance via punishment, which motivated our interest in performing similar analyses on sanction.

Villatoro et al. evaluate the cost-effectiveness of sanction against punishment. Their simulation randomly pairs agents as players of a Prisoner's Dilemma game. The agents are, again, allowed the binary option to comply or defect, and the agent's decision is persistent in subsequent rounds of the game. Agents that comply are able to sanction others in later stages to communicate norm compliance, whereas those that defect must use punishment. Villatoro et al. conclude that sanction is a much more cost-efficient means of garnering cooperation, which lends confidence to the relevance of our study.

In contrast to the aforementioned studies, our simulation recognizes sanction as the sole means for applying consequence to norm violation. This follows from our structured lab scenario where directed normative relationships exist from an adviser to his or her graduate student and from the network administrator to the students in the lab. Further, our simulation is not decentralized, as there exists a central authority who's responsible for all sanctions.

6. CONCLUSIONS AND FUTURE WORK

Using ENGMAS, our exploration of system resilience and liveness through variable observability and sanction type has yielded some interesting conjectures. For example, if the sanctioner has low observability, agents are less "controlled" and tend to violate security norms more frequently. As a result, their respective PCs enter a critical state, decreasing their utility in performing research tasks until they are let go from the organization. Student agents perform better under group sanction than under individual sanction in terms of total norm violations. Considering the time-cost associated with issuing sanctions, these results may suggest that it is more cost-effective to govern with group sanction, as the sanctioner may maintain security compliance with far fewer sanctions.

In other future work, we plan to further develop our scenario and its accompanying simulation in order to conduct more in-depth research on norm-governance in multiagent systems and we anticipate more variance on resilience. In-

corporation of multidisciplinary concepts from fields including anthropology, sociology, and psychology, among others, will help us to refine the behavior of our agents and increase the potential for more evaluative studies on norms and their properties. We further anticipate building a realistic model of agents with rational choices based on rewards, sanctions, and utilities, and their capabilities of observation and communication, and including cost of applying sanctions and the norm sanctioning capabilities of the governing principal to better portray the reality of implementing such policies.

Acknowledgments

We gratefully acknowledge National Security Agency for support via the Science of Security Lablet at North Carolina State University. We would like to thank Jon Stallings for helpful comments on a previous version.

REFERENCES

[1] R. M. Adams. Strategies of maximization, stability, and resilience in Mesopotamian society, settlement, and agriculture. *Proceedings of the American Philosophical Society*, 122(5):329–335, 1978.

[2] B. Alpern and F. B. Schneider. Defining liveness. *Information Processing Letters*, 21(4):181–185, 1985.

[3] R. Axelrod. An evolutionary approach to norms. *American Political Science Review*, 80(04):1095–1111, 1986.

[4] F. Berkes and N. Turner. Knowledge, learning and the evolution of conservation practice for social-ecological system resilience. *Human Ecology*, 34(4):479–494, 2006.

[5] M. Bishop. What is computer security? *IEEE Security Privacy*, 1(1):67–69, Jan 2003.

[6] R. L. Boring. Reconciling resilience with reliability: The complementary nature of resilience engineering and human reliability analysis. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 53, pages 1589–1593. Sage Publications, 2009.

[7] G. Dhillon and J. Backhouse. Technical opinion: Information system security management in the new millennium. *Communications of the ACM*, 43(7):125–128, July 2000.

[8] D. D. Heckathorn. Collective sanctions and compliance norms: A formal theory of group-mediated social control. *American Sociological Review*, 55(3):366–384, 1990.

[9] L. Lamport. Proving the correctness of multiprocess programs. *IEEE Transactions on Software Engineering*, SE-3(2):125–143, March 1977.

[10] S. Mahmoud, D. Villatoro, J. Keppens, and M. Luck. Optimised reputation-based adaptive punishment for limited observability. In *Self-Adaptive and Self-Organizing Systems (SASO), 2012 IEEE Sixth International Conference on*, pages 129–138, Sept 2012.

[11] P. Noriega, A. K. Chopra, N. Fornara, H. L. Cardoso, and M. P. Singh. Regulated MAS: Social Perspective. In G. Andrighetto, G. Governatori, P. Noriega, and L. W. N. van der Torre, editors, *Normative Multi-Agent Systems*, volume 4 of *Dagstuhl Follow-Ups*, pages 93–133. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2013.

[12] R. Plummer and D. Armitage. A resilience-based framework for evaluating adaptive co-management: Linking ecology, economics and society in a complex world. *Ecological Economics*, 61(1):62–74, 2007.

[13] T. B. Sheridan. Risk, human error, and system resilience: Fundamental ideas. (cover story). *Human Factors*, 50(3):418–426, 2008.

[14] M. P. Singh. Norms as a basis for governing sociotechnical systems. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(1):21:1–21:23, Dec. 2013.

[15] W. Vasconcelos, M. Kollingbaum, and T. Norman. Normative conflict resolution in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 19(2):124–152, 2009.

[16] D. Villatoro, G. Andrighetto, J. Sabater-Mir, and R. Conte. Dynamic sanctioning for robust and cost-efficient norm compliance. In *International Joint Conferences on Artificial Intelligence*, volume 11, pages 414–419, 2011.

APPENDIX

Table 7: Variables with Normal Distribution

Variable	Value
μ of Preference	0.5
σ of Preference	0.3
Upper Limit of Preference	0.8
Lower Limit of Preference	0.4
μ of Research Motivation	0.7
σ of Research Motivation	0.15
μ of Security Compliance	0.7
σ of Security Compliance	0.15
μ of Research Task Duration	5
σ of Research Task Duration	3
Lower Limit of Research Task Duration	1

Table 8: Experiment Parameters

Experiment Parameter	Value
Coefficient c for Research Task Duration	1.3
Naturally Decrease Rate of PC Health	0.1
Increase Rate of Agent Health	0.25
Decrease Rate of Agent Health	0.25
Increase Rate of PC Health	0.25
Decrease Rate of PC Health	0.25
Increase Rate of Research Motivation	0.25