Unbelievable Agents for Large Scale Security Simulation

Jerry Lin, Jim Blythe, Skyler Clark, Nima Davarpanah, Roger Hughston and Mike Zyda
University of Southern California

jerrylin@usc.edu, blythe@isi.edu, sclark@freolic.com, ndavarpa@usc.edu, rwhughston@gmail.com, zyda@usc.edu

Abstract

Human error arguably accounts for more than half of all security vulnerabilities, yet few frameworks for testing secure systems take human actions into account. We describe the design of an experimentation platform that models human behaviors through intelligent agents. Our agents share some desired features with believable agent systems, but believable interaction with a human is less important than accurate reproduction of security-related behaviors. We identify three main components of human behavior that are important in such a system: (1) models of emotion and other cognitive state that may increase the probability of errors, (2) flexible reasoning in the face of a compromised system and (3) realistic task-based patterns of communication among groups. We describe an agent framework that can support these behaviors and illustrate its principles with a scenario of an insider attack. We are beginning the implementation of the framework, and finish with a discussion of future work.

Introduction

Human error is widely recognized as one of the most important sources of vulnerability in a secure system. In a survey taken in 2006, approximately 60% of security breaches were attributed to human error by security managers (Crawford 06, Cranor 08). Humans often ignore or misunderstand warnings, underestimate danger, and download infected files or simply disable security mechanisms because of their slowness or complexity (Whitten and Tygar 99). Consider the old statement that the only secure computer is one that is turned off and/or disconnected from the network. A social engineering attack exploiting the human element would simply be to convince someone to plug it back in (Mitnick 02) .

But frailties are only one aspect of human behavior that impacts our understanding of security. Compared with software systems, humans are flexible and resourceful problem solvers, able to find alternate ways to accomplish their tasks despite failures of resources or services. Different people often perform the same task in different ways, providing a diversification defense from some attacks. Dourish and Redmiles (02) introduce the concept of "effective security" as a more realistic measure of the security of a system than a formal evaluation of the security mechanisms installed. The level of effective security is almost always below the level of theoretical security that is technically feasible in a system, largely due to human error. On the other hand, effective security must be measured end-to-end, taking into account the entirety of the system and the purpose it solves. In this context a high level of theoretical security may be both expensive and unnecessary.

Cranor (08) proposes a framework for reasoning about the security of systems with humans in the loop. She models the human as an information processor based on the warnings science literature (Wogalter 06). However, this model only captures the human response to warning messages and ignores many important aspects of human behavior, such as the task being performed, collaboration that leads to structured communication, and stress, emotions and tiredness that will affect a human's propensity to make errors. Cranor's approach allows a checklist-style evaluation of a security system.

In this paper we outline a research agenda to enable a more detailed and encompassing evaluation of human-in-the-loop security systems, using intelligent agents (Giampapa and Sycara 02, Chalupsky et al. 01). We are designing agents capable of simulating a shared task, in which individual agents have different roles, different basic skills and also different emotional responses. Such a framework should be able to answer far more detailed questions about the effective security of a system in a range of different scenarios. One of our goals is to capture those aspects of human nature that often prove to be crucial in the security of modern systems, for use in large-scale simulations at a level of fidelity that allows for end-to-end scientific evaluation.

Distribution Statement "A" (Approved for Public Release, Distribution Unlimited), DISTAR case #15423, 30 Apr 2010.

Given this goal, what aspects of human behavior are important to capture? We focus on three aspects of human behavior that have an important influence on the likelihood of success and severity of cyber attacks: (1) errors, particularly under time-related stress, (2) flexibility of response to problems and (3) non-random patterns of communication centered around a collaborative task.

While many believable agent-based simulation systems have been built (Bates 93, Choi et al 07, Marsella and Gratch 09), many of them are concerned with believability to humans through interaction (Tambe et al 95). None of these systems have the specific goal of capturing the human element that proves to be the weakness in computer security. Here, we are not so concerned with believable human-to-agent interaction, but in sufficiently similar action compared with human behavior to make simulation results valid. This is why we have used the term "unbelievable agents" to describe our approach.

Scenario

As we describe our agents' desired properties and architecture we will make reference to the following scenario: Three organizations are working on a joint project. Within their respective companies, there are team leaders, workers, and IT professionals. Each company may have a point of contact with the others and knowledge of how to communicate with other workers.

Two of the organizational teams gather information from different sources and primarily communicate between themselves. After gathering data, they update a cloud service spreadsheet with data they have collected and packaged for analysis. The third organizational team simply reviews the data, analyzes it, and updates with results.

There is a worker who is interested in infecting company computers for financial gain. He is a part of one of the teams and is aware of trust relationships within the work groups. He waits until another worker goes on lunch break and jumps on his computer, uses a password that was written on a post-it, and uploads a worm that propagates through email. An outsider coordinating with the malicious insider then gains access to information on various systems. At some point, a normal worker notices something is not right and contacts an IT worker he is familiar with. The IT workers attempt to coordinate and fix the issue.

Some security questions that may be answered through agent simulations are: What kinds of organizational structures are more resilient to cross-organizational attacks such as this one? What kind of policy is most effective? Was a piece of security hardware effective? How much of legitimate vs. malicious traffic is blocked by our security systems? How does this affect productivity? What kind of procedures can IT professionals take to mitigate damages once they are done?

Importance of Human Behavior in Security

Given that human frailties are an important aspect of computer security, to what degree do they need to be reproduced in software agents in order improve end-to-end evaluation?

To achieve our research goals, we need to model frailties in context of human-computer interaction . This does embody some understanding of how humans communicate, consume information, publish information and distribute information without a computer, but not the full scope of human behavior.

For experimenters, the ability to capture human behavior at different levels of fidelity is important. The benefit of accurately capturing the full range of human behavior on computers is clear. For partial capture of human behavior, we believe an experiment may want to focus on a specific phenomenon related to just a few human traits and capturing too much may add too much complexity and hinder analysis.

One of the open questions we aim to answer is whether there is an equivalent "uncanny valley" in simulating humans in such a manner. In other words, are there simulations which appear better but actually get worse results because we fall into specific errors near a good simulation?

Agent Properties

We divide the different properties we consider into properties of individual agents, and properties that govern patterns of communication within and between groups. Attributes of individuals are important to achieve a base level of fidelity as well as to provide a way to incorporate human frailties and behavioral diversity into the simulations. We will extend a standard BDI agent (beliefs, desires, intentions) (Bratman 87; Rao and Georgeff 91) in two main ways. First, we will incorporate modular goal-based planners. Second, we will integrate a cognitive/emotional state including several factors such as emotional response based on appraisal theory (Gratch Marsella 04), biorhythems (e.g. hunger, fatigue) focus level, stress, creativity/agility, and technical competency to adjust the planning and execution processes. For example, an agent who is more creative would be able to devise new plans to achieve their goals; or one who is fatigued and less technically competent might incorrectly override security mechanisms. These influences are dynamic. For example, as the simulation progresses, the agents will become more fatigued or if agents were given training, their technical competency could rise.

In order to model realistic patterns of communication, we will create and keep track of a social network for our agents. This will track whom they may be familiar with, the types of relationships they have, and their understanding of the other agents in the simulation. Agents can then reason about who they may think would be interested in a funny Youtube video or who they would contact first for help. In our scenario, a worker who suspects a worm would contact a person in IT he knows as

a friend, who then may be more inclined to listen and investigate rather than be annoyed. This aspect is important to simulate attacks that traverse a social network such as Facebook viruses or old Trojan viruses which used victim's instant messaging account to propagate. This happens in our example scenario where a worm propagates itself through email using the victim's address book.

When the network under attack contains an organization performing a task, as in our scenario, the needs of the task itself probably dominate the patterns of communication. In this case, one would expect denser communication within working groups than between them. The temporal pattern of communication will probably follow the working day and also the organization's deadlines. The social network is important within the organization for modeling leisure-related communication and determining who an agent is likely to approach for help with technical or security concerns.

Detailed tracking of an agent's social network is important for its emotional influence on decision making. For example, humans will sometimes avoid admitting fault to co-workers or superiors in an attempt to maintain the best possible relationship with others, because of pride. Alternatively, a person might also admit fault because of moral traits or feelings of guilt. Certain prejudices towards people such as those claiming to be from IT may also lead to an agent being more compliant to requests such as deleting files or turning on a machine.

The timing of agent behaviors is another important aspect for realistic simulations. Cyber attacks can take place over very short periods of time, far too quickly for humans to react. Our agents' decision-act cycles must match those of humans well enough to capture this. Similarly, changes to our agents' cognitive states, e.g. tiredness, hunger and frustration, should take place over reasonably human time scales.

Many of these desired properties are shared with agents that behave in a believable fashion in other domains, for example in games and for training, and we intend to make use of this work where possible. However the security domain makes some properties of believable agents almost irrelevant and other properties that have not been much studied are more important. For example, interaction with other humans is not, at least initially, a required aspect of believability in this domain, allowing us to finesse natural language understanding and generation or body language. There are also properties that we do not intend to model in the first version of our framework although they are important in the long run. These include the ability to learn from observations of the world and of each other, and the ability to influence the views and beliefs of another agent.

An interesting example of these differences is in the diversification of agents, *i.e.* to what extent different agents should perform the same tasks differently. In our domain, one way this is important is in how many individuals are vulnerable to an attack that relies on using a particular feature of some software that has a vulnerability. In the real world, it may be that a third of the users use this feature, while the rest perform the task in other ways. Without some diversification, all or none of the agents might be vulnerable to the attack. This can be contrasted with the game Halo, where players found the actions of the automated agents to be less believable if they were too varied. The designers made adjustments reducing the diversification of the agents.

Architecture and Implementation

Our agents are based on a well known BDI model, however we are extending it with what we call the agent's cognitive state. The cognitive state will influence normal intentions, goals, and possibly available actions and methods. Other modules such as planners, state analyzers, and learners may be integrated as plug-ins. This is also intended to allow for extensibility for specialized needs. The agent architecture is shown in figure 1.

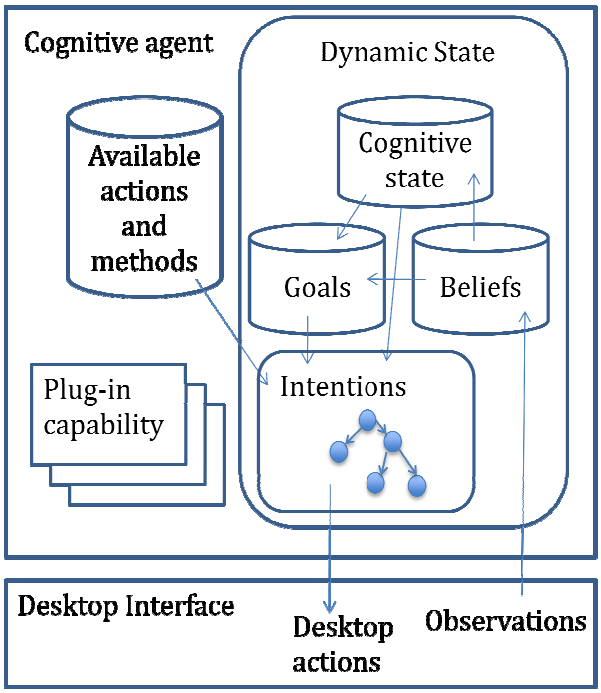

Figure 1.Our agent architecture is based on a BDI model with a cognitive state that includes emotions and aptitudes. A separate desktop interface provides an abstraction through which the agent interacts with its software environment.

Each agent is initialized with goals, beliefs, intentions, and a cognitive state; depending on the role that agent plays in the overall simulation. In the case of our scenario, the inside attacker's goals would be to make money, it has certain beliefs about information that flows through company machines and the technical competencies of its coworkers, and its intentions are to use a worm to gather

information for financial gain. With cognitive state, however, if the agent had sufficient laziness, for example, he may never follow through with his intentions or choose to pursue an easier path. It should be noted that these choices within the agent are stochastic and will rely on a psychological model of how different factors affect reasoning. This model may vary between agents to account for more flexibility and variance in behavior. The cognitive state is shown in figure 1.

Example attributes in the cognitive state include tiredness and stress level. As agents complete tasks without a break their tiredness increases. The stress level may increase if they notice evidence that the computer environment may be compromised, or if goals are obstructed. Elevated levels of tiredness and stress increase the probability that an agent will make mistakes, for example ignoring warning messages, or turning off security software to save time. The reason for this change in behavior is agent's reaction to their feelings and decision to cope with these feelings by becoming more careless. The need to cope with certain feelings may lead to other decisions such as taking breaks or giving up. Carelessness could be a primary reason the worm in our scenario goes unnoticed for a certain length of time. Eventually someone, perhaps from the IT group, will either suspect a strange email or notice strange system behavior.

For the actual implementation of our agents, we plan to leverage either Soar or SPARK (Morley and Myers 04). Soar is based on a unified cognitive architecture. In Soar, knowledge (actions and methods) is specified in a series of statements roughly in "if…then" form commonly seen in expert systems. Agents built on Soar have been shown to be very robust in the face of failure or uncertainty, which is important in our domain. Agents based on Soar are also capable of abstraction and learning from experience.

SPARK is a descendent of Georgeff et al.'s Procedural Reasoning System (Georgeff and Lansky 87) that was central in the development of BDI systems. SPARK is much smaller in scope than SOAR and concentrates on an efficient, flexible language for agent behaviors with a sound formal basis. Its representation for agent operators is more procedural, and similar to that of RAP systems (Firby 89). SPARK supports multiple execution threads for agents and the interruption and resumption of tasks.

Much of the reasoning about cognitive state concerns emotions. In common with several research groups, we view emotions as arising from goal achievement or failure and modifying the agent's actions. Relatively simple models have been implemented that have validity from cognitive science, for example Em (Neal Reilly 96), which is based on the models of Ortony et al (88) or EMA (Marsella and Gratch 09) based on appraisal theory (Smith and Lazarus 90).

Emotions in our agents are based on the work (Gratch Marsella 04) which is based on the appraisal variables of relevance, desirability, causal attribution, likelihood, unexpectedness, urgency, ego involvement, and coping potential. Every time an observation is made, appraisals are generated for these variables which contribute to the affective state, and lead to coping behavior or changes in cognitive state.

The flexible behavior required of our agents will be implemented through planning systems. Although both SOAR and SPARK are capable of simple planning, we anticipate the need for more sophisticated planning tools to operate quickly in large domains. These can be included as plug-in tools, as shown in Figure 1, where the agent will invoke a planner to help choose a next step, allowing the planner a filtered view of its beliefs and goals, and incorporating the result as intentions. For this reason we plan to use a Blackboard model for the agent's dynamic state (Engelmore and Morgan 86).

To work in teams, groups communicate in a hierarchical structure as shown in figure 2. Organizations are represented by agents who act as team leaders. The team leaders are assigned high level tasks or sets of tasks and decompose them into finer grain tasks which are delegated to team members. This not only reflects organizational structures in our scenario, but also many real human structures. Workers in the scenario will also rely on this organizational structure to collaboratively produce a spreadsheet.

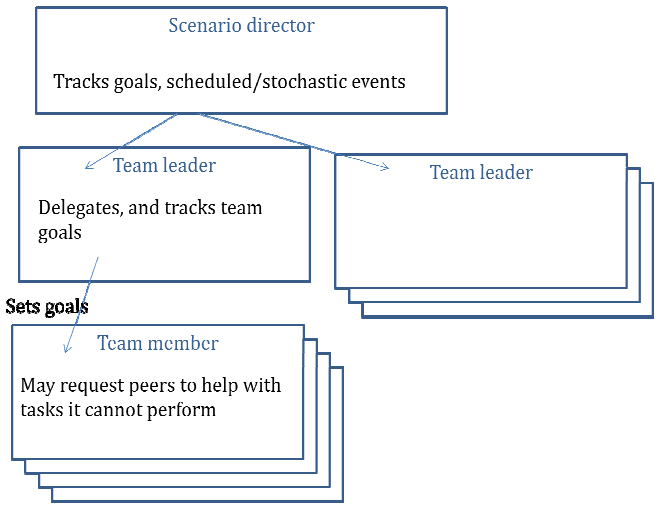

Figure 2. A Hierarchical agent communication framework is realistic and also supports scalability.

One agent is distinguished as the scenario director that keeps track of critical events that place as the scenario unfolds. The director maintains a high level view of the system's overall state, through communication with team leaders, and triggers critical events that move the scenario forwards at the appropriate moment. Actions that require agency are assigned to team leaders or key agents in the scenario. Examples of similar directors include Moe, part

of the Oz project, which used adversarial search to ensure that plot points were met while the user explored an interactive world (Weyhrauch 97, Kelso et al 93).
In our scenario, each of the three companies is created as a separate team, each with a team leader, workers, and IT professionals respectively. Two of the teams are designed to gather, collect and package data, and the other team reviews, analyzes, and updates with a final result. The outsider can be defined as his own team, but simply stays out of the story until the inside attacker is triggered by the scenario director. Most of the agents in this example have the ability to manipulate the data as a spreadsheet saved on a cloud service.

The inside attacker is given a special set of goals that can be triggered directly, and at a given time will compromise another agent's computer. This begins the attack phase of the simulation in which we can model and measure a number of features, including the size and speed of the attack, the amount of data compromised before IT professionals can re-secure, and the loss in overall productivity.

Conclusions and Future Work

Understanding the human element is critical in evaluating systems for security. We have outlined an architecture based on autonomous agents that will improve researchers' ability to incorporate human behavior into experiments with security systems. By allowing all the agents to be simulated, the approach maintains the benefits of automatic testing, such as scale and potentially accelerated timelines. We have also outlined a scenario in which team oriented behavior, human frailties and human flexibility of approaches play an important role and shown how it will be modeled within our framework.

We are currently in process of implementing a prototype of the framework. We intend to perform full evaluation on the system and improve upon our current design decisions. Two of our central questions moving forward will be scalability and user authoring of agents and behaviors.

We want to support experiments with perhaps thousands of agents performing loosely coupled tasks over a realistic hardware and network landscape. We believe our approach will scale, even with a single scenario director, if we allow the scenario director to offload the oversight of key plot events to team leaders as necessary. Experiments of security systems may take days or weeks to run, creating a challenge to the longevity of our autonomous agents.

In the long term we intend to construct a toolkit for security researchers, allowing them to instantiate human behaviors as appropriate for their experiment. This will rely on powerful authoring tools that will allow users to define the key plot points of a scenario and a set of agents, probably by retrieving agents from a library and modifying their capabilities and profile. We intend to build on earlier work in procedure editing already integrated with Spark as a starting point (Blythe 05).

Important questions remain about how to evaluate systems such as this and what conclusions can be drawn from experiments run with this system. Here we intend to follow work from other multi-agent or believable agent systems. Ultimately we aim to enable useful research making security tests that incorporate human error, the probable vulnerability point of more than half of all successful cyber attacks.

References

Engelmore, R., and Morgan, A. eds. 1986. *Blackboard Systems.* Reading, Mass.: Addison-Wesley.

Bates, J. 1993. The Nature of Character in Interactive Worlds and the Oz Project, *Virtual Realities: Anthology of Industry and Culture,* Loeffler, ed.

Blythe, J. 2005. Task Learning by Instruction in Tailor, *Intelligent User Interfaces*

Bratman, M. 1987. *Intentions, Plans, and Practical Reasoning,* U Chicago Press

Cranor, L. 2008. A Framework for Reasoning about the Human in the Loop, *Usability, Psychology and Security*

Chalupsky, H., Gil, Y., Knoblock, C., Lerman, K., Oh, J., Pynadath, D., Russ, T., Tambe, M. 2001, Electric Elves: Applying Agent Technology to Support Human Organizations, *Innovative Applications of Artificial Intelligence*.

Crawford, M. Whoops, Human Error Does it Again, *CSO Online,* http://www.csoonline.com.au/index.php/id ;25583 0211;fp;32768;fpid;20026681

Dourish, P. and Redmiles, D. 2002. An Approach to Usable Security Based on Event Monitoring and Visualization, *New Security Paradigms Workshop*.

Firby, J., 1989. *Adaptive Execution in Complex, Dynamic Worlds*, PhD Thesis, Yale.

Giampapa, J. and Sycara, K. 2002. Team-Oriented Agent Coordination in the RETSINA Multi-Agent System, *AAMAS 2002 Workshop on Teamwork and Coalition Formation*

Georgeff, M. and Lansky, A., 1987. Procedural Knowledge, *IEEE 74, 1383-1398*

Kelso, M., Weyhrauch, P. and Bates, J., 1993. Dramatic Presence, *PRESENCE: The Journal of Teleoperators and Virtual Environments, 2, 1*

Marsella, S. and Gratch, J., 2009. EMA: A Process Model of Appraisal Dynamics, *Journal of Cognitive Systems Research, 10, 1.*

Morley, D. and Myers, K. 2004. The SPARK Agent Framework, in *Intl Conf on Autonomous Agents and Multi-Agent Systems* (AAMAS 04)

Neal Reilly, S. 1996. *Believable Social and Emotional Agents*, PhD Thesis, CMU-CS-96-138.

Ortony, A., Clore, A. and Collins, G. 1988 *The Cognitive Structure of Emotions*, Cambridge University Press

Rao, A. and Georgeff, M., 1991, Modeling Rational Agents within a BDI Architecture, *Int Conf. on Knowledge Representation*

Smith, C. and Lazarus, R. 1990. Emotion and Adaptation. *Handbook of Personality: Theory and Research,* Guildford Press.

Weyhrauch, P. 1997. *Guiding Interactive Drama*, PhD Thesis, CMU-CS-97-109

Whitten, A. and Tygar, D. 1999. Why Johnny Can't Encrypt: A Usability Study of PGP 5.0, *Proc. 8th USENIX Security Symposium*

Wogalter, M. 2006. Communication-Human Information Processing (C-HIP) Model. In *Handbook of Warnings*, Lawrence-Erlbaum, NJ.

Mitnick, K. D., 2002 *The Art of Deception: Controlling the Human Element of Computer Security*, Hoboken, NJ: Wiley

Tambe, M., Johnson, W.L., Jones, R. M., Koss, F., Laird, J. E., Rosenbloom, P. S., Schwamb, K., 1995 Intelligent Agents for Interactive Simulation Environments, *AI Magazine* 16(1): 15

Gratch, J., Marsella, S., 2004. A Domain-independent Framework for Modeling Emotion, Journal of Cognitive Systems Research, 5, 4