EXPRESSING AN INFORMATION SECURITY POLICY WITHIN A SECURITY SIMULATION GAME

Cynthia E. Irvine and Michael F. Thompson

Naval Postgraduate School

Abstract: The Center for the Information Systems Studies and Research (CISR) at the Naval Postgraduate School has established a broad program in computer and network security education. The program, founded on a core in traditional computer science, is extended by a progression of specialized courses and a broad set of information assurance research projects. A CISR objective has been improvement of information assurance education and training for the U.S. military and government. Pursuant to that objective, CISR is developing a computer simulation game, CyberCIEGE, to teach computer security principles. CyberCIEGE players construct computer networks and make choices affecting the ability of these networks and the game's virtual users to protect valuable assets from attack by both vandals and well-motivated professionals [1]. CyberCIEGE includes a language for expressing different security related scenarios. A central part of this language is an ability to express a variety of different information security policies.

Key words: Information Assurance, Security Policy, Simulation Game, Scenario Definition Language

1. INTRODUCTION

Computer and network security is a broad field with many subtle properties, not the least of which is its role as a negative requirement. In other words, the issue is often what does not happen, e.g., a secret asset should not be disclosed to an enemy. Educating students to appreciate the subtle elements of computer and network security is greatly enhanced by a tangible context in which cause and effect can be experienced. One means of providing this context is the use of labs containing networked computers and exercises illustrating specific attacks and defenses. Use of labs to provide context to lessons is limited to those students with access to the physical lab equipment. Also, students typically perceive a lab exercise as just that: an exercise. It is difficult for students to have a sense of what it is they are protecting and who or what they are protecting it from.

Security choices are often a risk management tradeoff between threats and the costs of deploying protection mechanisms, including detrimental effects on productivity resulting from security choices. Unless the student has a sense of the value of what is protected, and an understanding of the strength of motive of potential attackers, it is hard to gauge the required strength and assurance of protection mechanisms. Similarly, it is hard to appreciate the effects of

security choices on costs and productivity unless there is a constrained budget and users who must directly or indirectly interact with the protection mechanisms to productively utilize the computing resources. CyberCIEGE is designed to immerse the player in this risk management context. Player choices have implications for both the protection of information, and costs to a virtual enterprise.

2. A LANGUAGE FOR EXPRESSING A RANGE OF SCENARIOS

Early in the project we concluded that a range of different risk management contexts are needed to illustrate how the effects of security choices depend on the security policy and the physical environment. To achieve this, we defined a language with which to express the security related risk management tradeoffs for different scenario contexts, which we call "scenarios". The CyberCIEGE game engine interprets this scenario definition language and presents the player with the resulting game. What the player experiences in a given instance of the game and the consequences of the player choices are a function of the scenario as expressed using the scenario definition language.

2.1 Security Policy

The question of whether a system is secure depends in large part on the security policy. Stated another way, a system can only be said to be secure with respect to some well-formed policy expressed in terms of information and authorized user access to information. The CyberCIEGE scenario definition language captures the abstraction of information security policy through a construct called an "Asset", which represents information of some value to the enterprise. The scenario designer defines multiple assets, each having a set of attributes (e.g., the value of the asset, who is authorized to access the asset, etc.). Taken together, these asset definitions reflect the abstract information security policy for the enterprise modeled within the scenario.

Expressing an abstract information security policy requires identification of what is being protected from whom. When an asset is defined in the language the core notion of "what" is expressed as a cost to the enterprise in the event of some form of compromise. For example, "the enterprise would loose the equivalent of $500,000 should the secrecy of this asset be compromised." Or, "the enterprise would loose the equivalent of $100,000 should the integrity of this asset be compromised". Similarly, the notion of "whom" is expressed as a motive for some form of compromise. For example, "an attacker has a very high motive to succeed in compromising the secrecy of this asset.

The language represents loss to the enterprise in terms of dollars, which is the metric by which the player's success is measured. The scenario designer controls the initial amount of money available to the player, and a monthly budget that fluctuates based on the productivity of the virtual users within the game. The scenario designer also controls the monetary loss resulting from compromise of assets. Thus, the scenario designer determines the impact of any given compromise on the success or failure of the player. Some loses can result in annoyance. Others bankrupt the entire enterprise.

Motive is expressed as a value ranging from zero to one thousand. Motive represents the lengths to which an attacker will go to compromise the asset. Table 1 summarizes motive values as they are used in the game. The higher the motive, the greater the protections needed to prevent compromise of assets. Motives of 800 and higher result in professional attacks utilizing subversion [2].

Table 1. CyberCIEGE Attacker Motives

Motive Value	Description
1000	Unstoppable, you trust no one.
800	Major interest to professional attacker
400	Strong interest to professional attacker
100	Minor interest to professional attacker, major interest to skilled hacker.
50	Minor interest to skilled hacker, major interest to unskilled hacker
25	Minor interest to unskilled hacker
0	Nobody will even read your blog.

2.2 Expressing Different Kinds of Policies

There may be different motives for compromising the same asset, with differing consequences. For example detailed test data for a secret formula might be of great value to a competitor, with catastrophic consequences to the enterprise from its compromise. On the other hand, a rival engineering manager within the same enterprise may be motivated to gain access to the same test data for purposes of showing how far behind the project is, with the intent of taking it over. To reflect these differing motives and costs, each asset has an associated "cost list" reflecting potentially different costs and motives for different "attackers". Cost lists also can distinguish between different modes of compromise, permitting the scenario designer to differentiate between damage resulting from disclosure and damage resulting from unauthorized modification.

Additionally, each asset may also have a secrecy label and/or integrity label to reflect labeled information security policies. These policies are sometimes called "mandatory access control (MAC) policies"[3]. The asset label attributes name "secrecy" and "integrity" constructs that are also part of the scenario definition language. For example, the scenario designer can define a secrecy level of "Secret, and give it attributes reflecting a loss to the enterprise resulting from disclosure and a motive for attackers to disclose any asset with that label.

Within scenarios that include labeled assets, the costs and motives associated with the "enemy" (e.g., a ruthless competitor) are reflected in the secrecy and integrity labels. And the costs and motives associated with internal intrigues, rivalries, ineptness and informal "need-to-know" are reflected in the explicit cost lists associated with each asset. In traditional computer security jargon, the cost lists are intended to reflect discretionary access control (DAC) policies while the labels reflect mandatory access control policies.

In addition to cost-list and label-based costs and motives, each asset includes an attribute reflecting the damage to the enterprise should the asset become unavailable due to a denial of service attack. Assets also have a corresponding denial of service motive.

Potential attackers (i.e., those motivated to compromise assets) include unauthorized enterprise users named in asset cost lists, and "external" attackers. Within the game, the means employed by attackers to compromise assets is a function of their motive. And, with suitable motive, external attackers will bribe internal enterprise users (i.e., the veritable "insiders") to compromise an asset. This includes users who are authorized to access an asset as well as unauthorized users who may have to defeat protection mechanisms to compromise the asset. Thus, CyberCIEGE represents "insiders" in two ways: those named in asset cost lists with explicit motives to compromise assets for which they are not authorized, and those who are bribed or otherwise coerced by external attackers.

2.3 User Trustworthiness and Insider Attacks

No amount of technology-based protection mechanisms can defend against an insider who can be bribed into compromising assets for which the insider is authorized. The virtual users within CyberCIEGE have a scenario-defined "trustworthiness" and a player-selectable "background check". The player can purchase different degrees of background checks for different groupings of virtual users, thereby reducing the vulnerability of users to bribes. For example, those users who are authorized to view valuable secrets can be given high background checks while other users have low or no background checks. As a resource, background checks are relatively expensive, so the player is dissuaded from purchasing background checks for all users who might gain access to the sensitive assets.

The modeling of background checks within CyberCIEGE is not entirely consistent with real world behavior. This results from a few bounds placed on the game development intended to limit complexity of game play. For example, the virtual users are static in each scenario. The player does not hire or fire users (though the player does hire and fire IT support staff and guards). Also, the player does not assign work to useres. The scenario designer defines which assets each user must access to perform work. Thus, if a user is not trustworthy, the player can't fire the user or assign the user to low risk assets. However, the player can purchase background checks. Doing so magically improves the user trustworthiness – but at a monetary cost. This level of modeling is sufficient to illustrate the value and the cost of background checks.

3. ENFORCING THE POLICY WHILE GETTING WORK DONE

To succeed in the game, the player assesses the information security policy of a given CyberCIEGE scenario by understanding:

1) The trustworthiness and asset authorizations of the users; and,
2) The costs to the enterprise resulting from asset compromise and the motive of attackers to achieve compromise.

The other half of the risk management tradeoff is the need for users to access different assets to do their jobs. CyberCIEGE includes a construct called an "asset goal", which the scenario designer uses to reflect productivity losses resulting from an inability of users to access assets needed to be productive. Within the game, a given asset is in one of two places: either in a user's head or on a computer component (e.g., workstation or server). Users can only achieve asset goals if the assets are on computer components. Thus, the player must provide the users with computer components with which to process the assets.

3.1 Productivity Requires Assets on Computers

The game includes a catalogue of computer components that player can purchase to satisfy the virtual user needs to process assets. Some scenarios start with components that initially contain assets. Other scenarios might start with no components, requiring the player to purchase and configure components. Figure 1 illustrates a CyberCIEGE virtual user who has been provided a workstation with which to access assets.

Figure 1. CyberCIEGE User Productively at Work

Many users may need to access the same asset to achieve their respective goals. This drives the player to connect computer components together using networks. Players can select from multiple different networks, including the Internet. The player can purchase network devices such as routers and firewalls to interconnect networks.

Each asset is instantiated on at most one component. Left to their own devices, users will store assets on components that are most efficient for all the users to access. The player must make choices to constrain which assets are processed on which computers.

Each user can have multiple asset goals. Each goal identifies one or more assets that must be accessed by the user to achieve the goal. When multiple assets are named in the same goal, the user must be able to concurrently access the assets from the same workstation. This allows the scenario designer to require the player to achieve a form of controlled sharing of assets across a range of security labels, resulting in a need for processing in a multilevel mode [3].

Optionally, asset goals also identify software applications that must be used when achieving the goal (e.g., require the use of an email client to access email assets). CyberCIEGE includes a catalogue of software applications, some of which are only available on certain operating systems. In some cases, there are multiple products of the same type of application, e.g., different email clients. And some products are vandalized more frequently than others. Also, to support information integrity scenarios, some applications and operating systems have more integrity than others. This can be used to illustrate the risks of low integrity software accessing high integrity assets [4].

Scenarios can be constructed such that failure to achieve one asset goal may be very costly to the enterprise, while failure to achieve a different goal results only in an unhappy user. Each asset goal includes a productivity attribute and a happiness attribute. The scenario designer can introduce goals that encourage the player to make unwise security choices in response to user

complaints. In such a scenario, the player may have to live with unhappy users to maintain security.

Asset goals can be dynamically introduced into a scenario based on current game conditions, e.g., the passing of time or a degree of success by the player. Altering asset goals is the primary means by which the scenario designer engages the player in an ongoing narrative. These changed goals can lead the player to buy new components or software, or make new network connections such as risky connections to the Internet.

3.2 Protecting Assets on Computers

Components have different security properties, e.g., operating systems that enforce policies with differing levels of assurance. Also, the player can choose to alter the security configuration of components. For example, workstations can be configured to automatically log off users after a period of inactivity. And players can set procedural security policies such as prohibitions against writing down passwords and the frequency of anti-virus updates. These of course depend to a great extent on user training, which is another set of choices the player can make. The variety of choices a player can make that potentially affect the security of components is shown in the screen capture of figure 2.

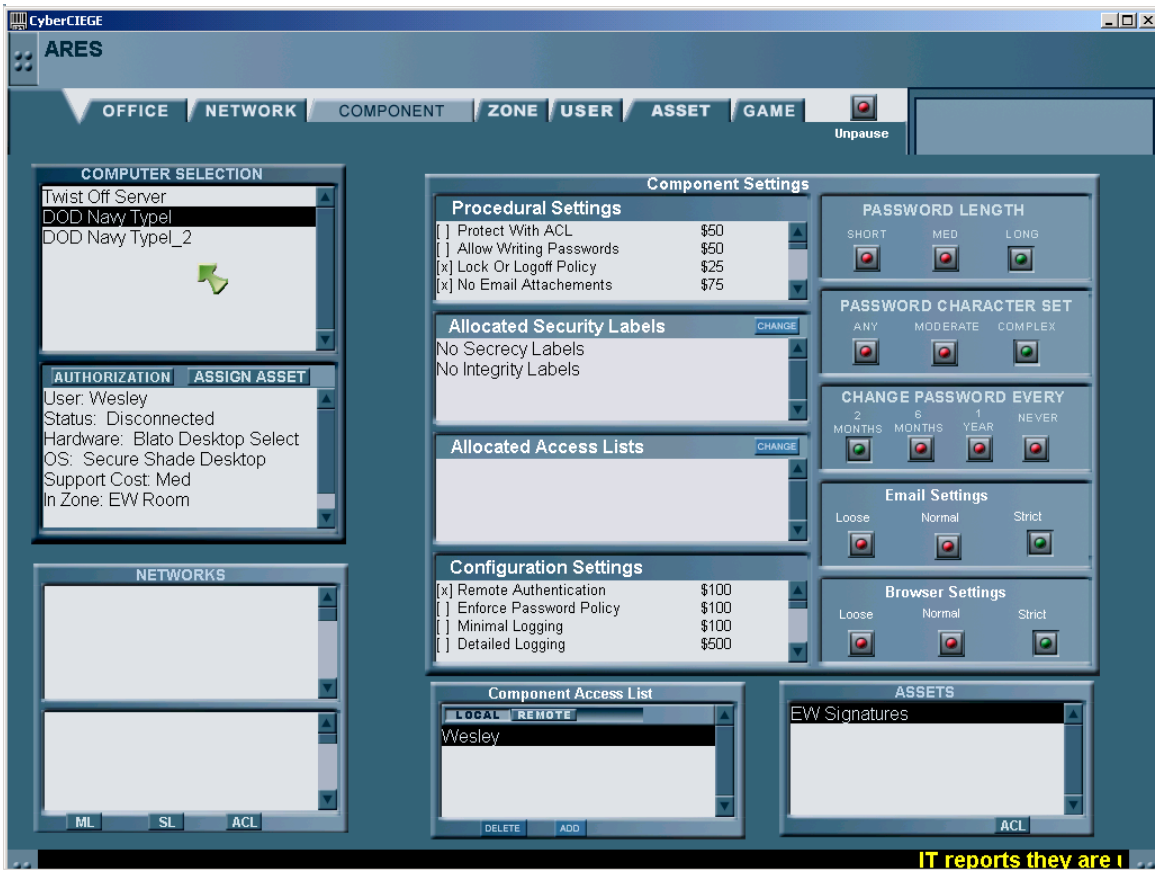

Figure 2. CyberCIEGE Component Configuration Settings

The operating systems on CyberCIEGE components affect the security of the component and constrain the kinds of applications available for use on the component (e.g., may limit what

applications can run on a specific server). Most operating systems can be configured to enforce a DAC policy. Some can be configured to enforce a MAC policy. And some operating systems have more assurance than others. The MAC operating systems include multilevel network connections and single level network connections. The player is responsible for selecting the security attributes of network connections to components.

Components include local and remote user accounts, allowing the player to constrain which users can access which components. Of course the enforcement of these constraints depends on factors such as operating system assurance and IT support staff.

3.3 Physical Protection of Assets

The extent to which computer component protection mechanisms are relied upon to enforce the security policy depends in large part on the degree of physical security. In some environments it is possible to store assets on components having very weak security, e.g., if the component is in a locked vault with no network connections. Each game scenario includes one or more physical zones that can be used to control the physical movement of users. An example of a zone is a physically secure office with a locked door for which only selected users have a key. When components are purchased, they are placed within a specific zone. Physical access to components can therefore be constrained based on the physical access to the zone. Players can increase the physical security of a zone by purchasing items such as guards, cipher locks and alarms. Players also choose who is permitted to enter the zone, and place constraints on what a user can carry into and out of the zone (e.g., cell phones and cameras).

Networks can extend between zones, resulting in an opportunity for attackers to engage in wiretap attacks. CyberCIEGE components include VPN gateways and link encryption devices that players can deploy to protect against such attacks.

4. CONCLUSION

The CyberCIEGE scenario definition language can be used to express an information security policy and define a work environment where user access to assets is necessary to maintain productivity. This allows a scenario designer to force the player to address the fundamental tension of computer security: Provide users with suitable resources to productively perform work, while protecting the assets from security compromises in accordance with the enterprise security policy.

REFERENCES

[1] Irvine, C. E., and Thompson, M., Teaching Objectives of a Simulation Game for Computer Security, *Proceedings of Informing Science and Information Technology Joint Conference*, Pori, Finland, June 2003.

[2] Anderson, E. (2002, March). A Demonstration of the Subversion Threat: Facing a Critical Responsibility in the Defense of Cyberspace. Masters Thesis. Monterey, CA: Naval Postgraduate School.

[3] Brinkley, D.L. and Schell, R.R. (1995). Concepts and Terminology for Computer Security. Information Security. ed. Abrams, Jajodia, and Podell. Los Alamitos: IEEE Computer Society Press. Retrieved November 21, 2002 from World Wide Web http://www.acsac.org/secshelf/book001/02.pdf

[4] Irvine, C., and Levin, T., "A Cautionary Note Regarding the Data Integrity Capacity of Certain Secure Systems," in Proceedings of the Working Conference on Integrity and Internal Control, Brussels, Belgium, 15 November 2001.