Teaching Objectives of a Simulation Game for Computer Security

Cynthia E. Irvine and Michael Thompson
Center for the Information Systems Studies and Research,
Naval Postgraduate School, Monterey, USA

irvine@nps.navy.mil **mfthomps@nps.navy.mil**

Abstract

This paper describes a computer simulation game being developed to teach computer security principles. The player of the game constructs computer networks and makes 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. The game introduces the player to the need for well formed information security policies, allowing the player to deploy a variety of means to enforce security policies, including authentication, audit and access controls. The game will depict a number of vulnerabilities ranging from trivial passwords to trap doors planted by highly skilled, well-funded adversaries.

Keywords: Computer Security, Education, Simulation, Game

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
27 JUN 2003	**N/A**	**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Teaching Objectives of a Simulation Game for Computer Security	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Postgraduate School 833 Dyer Rd Monterey, CA 93943-5118**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**UU**	**15**	

Introduction

The last several years have seen a tremendous increase in awareness of computer security threats and countermeasures. Ten years ago, if a mainstream periodical ran an article that mentioned computer vulnerabilities the novelty of making the news would create a stir in the computer security community. Today, grandmothers wonder if computer viruses are the source of their problems with "Web TV". Despite this increased awareness, users continue to select trivial passwords and network administrators routinely deploy systems without configuring them to reduce the risks of their being hijacked by children as platforms for engaging in vandalism against other systems. Further still, corporate and government policy makers often elect to deploy weak protection mechanisms in environments subject to potentially highly motivated hostile attacks. For example, the US General Accounting Office has again issued a failing grade to the security of most US Government computer systems (GAO, 2002). The increased awareness of computer security does not appear to translate into significant changes in behavior. The assumption in the work described in this paper is that much of the increased awareness of computer security is in the realm of hype and myth while in the real world most people don't give a second thought to less newsworthy yet consequential concerns. Examples of the latter are trivial password choices (CEO's and government leaders are notorious for these) and management decisions to connect critical networks to potentially hostile networks without a sound technical basis for trusting the associated protection mechanisms.

Education and training in computer security is often mundane and boring for both users and administrators. And some of the more critical conceptual issues are subtle, often eluding policy makers whose perceptions have been shaped by sensationalized accounts of "good hackers" and "bad hackers" engaged in a perpetual game of cat and mouse. A considerable amount of useful information has been published about the fundamental concepts of computer security (Brinkley, 1995; Pfleeger, 2003; Summers, 1997; Bishop, 2003), however sometimes people have to experience a problem in order to understand it. In addition, like many forms of engineering and medicine, the practice of computer security requires knowledge of the requisite facts as well as a tacit understanding of the art of security engineering. Computer security training and education can greatly benefit from an engaging means of presenting material that can potentially alter behavior of a broad audience including typical users and policy makers.

The remainder of this paper describes work to construct a commercial-quality game for teaching computer security concepts. The next section provides additional background regarding the educational program at the Naval Postgraduate School and motivation for the game. This is followed by a description of the elements of the game's underlying simulation and the way it will be played. We then discuss the information assurance concepts to be conveyed by the game. The worldviews and strategies applied in the game are presented followed by a summary.

A Game to Illustrate the Effects of Security Choices

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 (Irvine, 1997; Irvine 2001). The program is founded on a core in traditional computer science that is extended by a progression of specialized courses and a broad set of research projects in information assurance. An objective of CISR has been improvement of information assurance education and training for the U.S. military and government.

Better materials are needed to convey the general concepts of computer and network security to a broad audience unaware of the routine measures that could be taken to improve the protection of sensitive and critical information. In addition, the combined tension between the need for continued higher education combined with a widely distributed work force presents challenges in the delivery of pedagogical material. CISR has sought ways to extend its educational program beyond the physical boundaries of the campus.

Training and education in information assurance can take many forms. A challenge is to create a tool that not only conveys information, but also teaches users how to apply that information in a variety of situations. Several choices are available:

- A homogeneously trained pool of computer network security personnel can be created. Unfortunately, this uniform target will make the adversary's job easier, since personnel would provide merely rote responses and can be expected to react similarly to an attack and to apply lowest-common-denominator security engineering techniques to system design and implementation.
- It can provide distributed management of educational content, depending upon local organizations to provide training. This will provide heterogeneity, however, there is a danger that some localities may be very weak.
- It can provide a set of uniform training on tools and techniques, and simultaneously build a cadre of personnel who understand what they are doing and have the background and concepts needed to adapt to changing conditions and apply internalized concepts to new architectures. This third approach has the salubrious effect of making systems more difficult targets to attack.

The third approach has been chosen and, as part of that effort, we are building a computer game based on computer security concepts. This strategy for enhancing CISR's ability to provide computer and network security education exploits an established market and the growing acceptance of computer games in education (Kirriemuir, 2002). The game will simulate a range of scenarios involving computer networks. An example scenario is a business enterprise (e.g., a pharmaceutical company) that possesses highly valuable assets within a computer network. In such a scenario, employees of the enterprise require on-line access to a variety of assets having different values, yet the assets must be protected from compromise. The game lies in the tension created by the competing goals of efficient and affordable access to assets and protection of assets from unauthorized disclosure or modification.

The security simulation game will allow players to assume one of two distinct game player roles: defender or attacker. The player assumes one role, and the computer (or eventually another player) will assume the other role. As the defender, the player will construct computer networks having components (e.g., workstations, servers, etc.) that contain assets of differing values. The player will make security-relevant decisions about the network components and their interconnections. The player will also make decisions that affect the behavior of a set of virtual "user" characters that perform other roles within the enterprise and must efficiently perform work for the player to succeed at the game. Hostile game characters include vandals, insiders (e.g., disgruntled users), incompetent users and professional attackers willing to make significant investments to compromise high-value assets. Attackers will employ technical exploits as well as social engineering. As an attacker, the player will assess the vulnerabilities of specific computer networks within the context of specific physical conditions, and then plan and execute attacks.

For example, a player might exploit a software flaw to install a "trap door" that allows continued access the system even after the flaw is "patched".

Such A Game Exists: It is the Internet

A flippant response to this description of a computer security simulation game is to observe that the game already exists in the form of the Internet. One need only hook a network to the Internet and many parts of the "defender" role are suddenly thrust upon you. Vandals appear out of thin air, well prepared to provide you with a few lessons in computer security. Similarly, the Internet offers ample opportunities to those who would prefer to play the role of the attacker. Besides the obvious legal and moral deterrents to treating other peoples' systems as a part of some large computer game, the "Internet as a game" approach to training and education has other limitations. The greatest limitation concerns the matter of motive, which very much drives the means by which an attacker will attempt to compromise assets, and correspondingly the types of protections that are effective. A currently popular exercise in the computer security field is the creation of *honey pots,* which are computer networks placed on the Internet for the purpose of observing the techniques employed by attackers. The theory is that by deploying various protection mechanisms as part of these honey pots, you can learn which defenses are effective. However, unless information assets with an equivalent value of a million dollars are managed on the honey pot, it is unlikely that the full range of available lessons in computer security will be experienced. Honey pots attract the adventurous and vandals. They are less likely to attract well-motivated professionals who have conducted background research sufficient to indicate the value of information assets located within the honey pot. Thus these devices fail to illustrate attacks against high-value assets and do not illustrate to defenders the consequences of major losses.

A simulation game allows the player to experience the effects of security decisions (e.g., watch as the player-created virtual enterprise is looted because its "users" select trivial passwords) without taking the risks necessary to experience real loss, and without violating laws or established norms of behavior. A game can be configured to allow players to experience a wide variety of scenarios, and can be tuned by educators to complement lectures and illustrate specific points and concepts. An important question to be answered is whether the significant investment in a game will result in a high payoff in education, training, and awareness regarding information assurance concepts.

Can A Computer Security Game Be Fun and Educate?

Research shows the value of play in learning environments for children and adults (Reiber, 1996). However, when confronted with the idea of a computer game based on computer security concepts, the typical response from computer security professions suggested that such a thing would not be interesting. How can you make a fun game out of passwords, encryption and "access control lists"? On the other hand, computer game developers reacted quite differently. They observed that if a game premised on the administration of a virtual amusement park (Ham, 2000) can become a best seller, then a virtual world of corporate spies, obnoxious vandals, trap doors and Trojan horses could well be compelling to game players. The game developers suggested that the security game take the form of a resource management game in which the player starts the game with some set of resources including a finite budget. In such a game, the player constructs a network to support an ever-growing enterprise, reaping the benefits of productive users, expending resources to make the users happy while balancing benefits of protecting their

assets against the risks of losing resources (e.g., time and money) to vandals and professional attackers.

Games are attractive because they challenge players, require the use of imagination, and satisfy the player‘s curiosity, thereby encouraging experiential and exploratory learning. The pedagogical advantages of games include their ability to motivate students and as a vehicle for conveying a large body of information (Kirriemuir, 2002).

Elements of the Game

At the heart of the game are simulations that include several components: networks of interconnected components; virtual users attempting to accomplish work; and attackers who exploit vulnerabilities and weaknesses of the users (e.g., through “social engineering”). An overview illustration of the relationship between the network simulation and other game elements is illustrated in Figure 1.

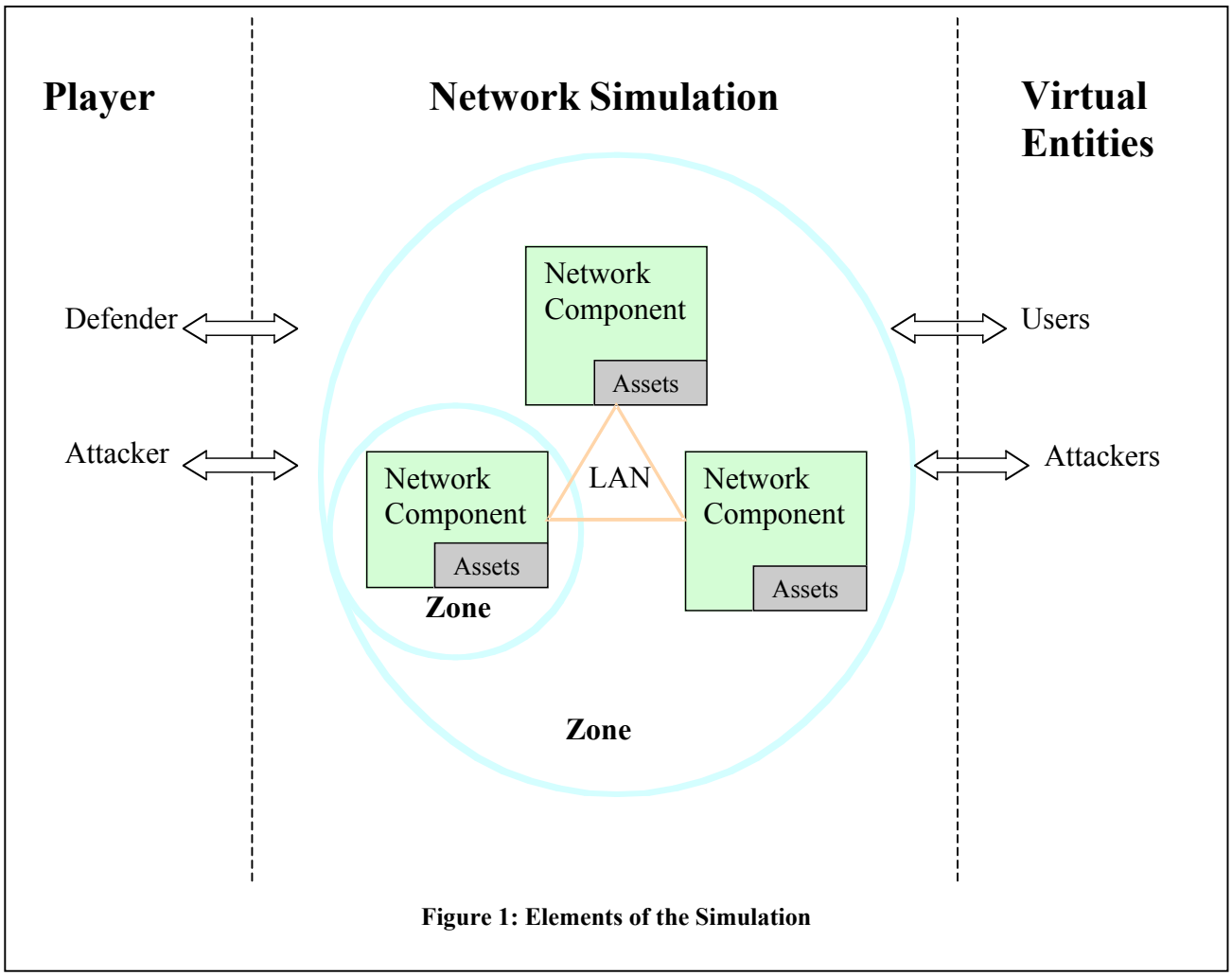

Figure 1: Elements of the Simulation

Constructing a Network

One goal of the game is to provide the player with some ability to engage in network construction, e.g., selecting and deploying new network components. Optimally, the game architecture will permit an evolution of the game implementation over time to support construction with components having less and less abstraction. Initially though, a fairly abstract set of components is defined with future extensibility built into the underlying architecture. Components have security properties and abstract functional properties. Security properties will include vulnerabilities such as flaws, Trojan horses and trap doors.

The player constructs the virtual network from network components; wire plants, and physical boundaries. Network components include workstations, servers and specialized components (e.g., firewalls, VPN gateways, etc.). Wire-plants include Ethernet LANs, the Internet, dedicated communication lines, etc. Servers include general-purpose file servers, e-mail servers and web servers.

Network components (e.g., workstations and servers) contain assets (viz., information having value). The network components also include applications that access the assets on behalf of users. Some applications include simulated human interfaces that can be invoked by the virtual users (e.g. the player's virtual "boss" or other "users"). In addition to simulated applications, network components include simulated operating system functions and mechanisms for enforcing security policies. All of these simulated software elements can potentially contain flaws or malicious software such as trap doors and Trojan horses.

Physical boundaries are called "zones". At any instant, each network component exists in one and only one zone. Wire plants (e.g., Ethernet LANs) can extend to multiple zones. Virtual users can enter and exit zones. Users can only physically access network components within the zone the user currently inhabits.

Keeping Virtual Users Happy and Productive

The network is constructed for the benefit of virtual users who utilize it to store and access assets having different values.

During the game, virtual user entities will interact with the simulation. For example, users try to accomplish work (e.g., access assets necessary to do a particular job). Users entities also engage in behavior that affects the security of the network (e.g., choose poor passwords, unwittingly support social engineering attacks, introduce potentially malicious software into the system, etc.) When the game player has assumed the role of defender, virtual user entities can nag, harass and whine to the player with demands for more connectivity, more real-time data sharing and more applications.

Game Characters: Users and Attackers

The behavior of the game's various virtual characters is defined by their goals. The following list enumerates the different character roles within the game.

- **Typical User** -- Just wants to do the job, and not trusted to handle sensitive information. Dislikes hassles. Generally does a good job and does not access information that is not part of the basic job description. Willing to learn new things (e.g., via training), but will

find an expedient (and potentially damaging) work-around if the "right way" is not obvious and easy.

- **Trusted User** -- Generally trusted to handle sensitive information. Overall goals give this user a lot more tolerance for hassles and as willingness to learn what is needed. May need to read and modify high sensitivity data at the same time as reading low sensitivity data (e.g., from the Internet). Optimally, both types of data will be on the same physical system, permitting combined real-time processing. At a minimum, this user desires a scheme to get copies of low sensitivity data onto the physical system that processes high sensitivity data.
- **Angry User** -- Looking for ways to harm the enterprise, but the motive is spite and the user would avoid punishment in most situations. Not motivated to either expend a lot of energy to achieve damage, or to take risks that would result in getting caught and embarrassed or fired.
- **Aggressive Incompetent User** -- Not authorized to handle sensitive data, and generally will not violate a clear policy (e.g., a mandatory access control policy). However, this user's eagerness will cause the user to stretch the meaning of "discretionary" policies. The biggest problem with this user is ignorance of the user's own limitations. This user may cause damage to a lot of information that the user should not even have access to.
- **Vandal** -- Motivated by boredom, desire for attention and/or just plain technical curiosity. Not motivated to expend significant resources, but is technically capable (e.g., could author viruses or install software onto a machine to which access has been gained). Some of these characters are deliberately obnoxious and attempt to explicitly damage resources. Others simply wish to experiment and do not view their own actions as malicious, however their mere entrance into the system costs the player resources (e.g., to perform forensics to ensure no damage was done).
- **Professional Attacker** -- Motive largely driven by resource values. Often interested in clandestine attacks that cannot be traced back to the attacker's organization. Able and willing to dedicate both time and funds on the attack, e.g. to send agents to go to work for your enterprise. Getting an agent installed as one of your "trusted users" would be a big challenge, but getting an agent (or corrupting an existing user) as one of your typical users would be relatively easy. The professional is also willing to send agents to go to work for your software vendors with a goal of installing a trap door or Trojan horse into the software you use, and thus subvert your system.

Destroying What You Have Built: Become the Attacker

As a defender, the primary goal of the player is to cause the simulation to reflect effects of choices on network vulnerabilities and the ability of virtual users to accomplish work. The game allows its players to see the network from antoher perspective: the player will be able to assume the role of an attacker; potentially attacking the very network constructed and defended earlier in the game. The attacker role will illustrate the effectiveness of different types of attacks against a network having various kinds of defenses. As the attacker, the player will make decisions regarding tactics (e.g., install a Trojan horse for later exploit) and will see the effectiveness (or lack thereof) of the network defenses.

What Are the Lessons to Be Learned?

The game is intended to provide training and education to a fairly broad target audience. Simple lessons such as the value of non-trivial passwords and the risks of walking away from a workstation without first logging off are readily provided to any player of the game. Some of the more subtle aspects of computer security, such as the inability of firewalls to effectively constrain information flow, require the use of more sophisticated scenarios. Because future computer security professionals are an important target audience of the game, a series of scenarios have been constructed to introduce this audience to the set of lessons enumerated below.

An Enumeration of Lessons

1) Computer systems (including networks of computers) can only be said to be "secure" with respect to some defined "information security policy". Therefore if those responsible for the security of the computer system do not know the information security policy, then the computer system, despite the presence of a variety of security features, is not likely to be secure.

2) An "information security policy" is not a technical or engineering document, and it is not an implementation document. Rather, an information security policy is typically a management directive that identifies the sensitivities of information and the constraints placed on people who might have access to the information (Sterne, 1990).

3) Historically, there are three kinds of information security policies:

a. Confidentiality – prevention of the unauthorized disclosure of information

b. Integrity – prevention of the unauthorized modification of information

c. Availability – prevention of the unauthorized withholding of information or resources. (This includes the unauthorized theft of computational resources since it fundamentally results in the withholding of the stolen resources from those who are paying to use them.)

4) Equal to the importance of knowing the information security policies is having assurance that the policies are in fact enforced. Dependence on technology to enforce an information security policy requires a defined level of assurance. Misplaced confidence in the security of a system is worse than having no confidence at all in its security. For example, consider the misplaced trust in a supposedly "secure" cryptographic system, such as Enigma, that was a serious contributor to the loss of World War II by the Axis powers (Kahn, 1967; Winterbothum, 1974).

5) The ability to achieve a high level of assurance that a system enforces confidentiality or integrity policies is different in kind from the challenges associated with availability policies. In general, the subjective nature of availability renders it impossible to verify to a high degree of assurance that a system possesses the quality of availability, particularly in the face of malicious software

6) Enforcement of confidentiality and integrity policies can be verified to a high degree of assurance. Enforcement of a subset of policies known as "non-discretionary" or "mandatory" policies is verifiable even in the face of malicious software. Non-discretionary policies are characterized as having precise definitions, and are both global and persistent.

7) High degrees of assurance cannot be achieved through testing and cycles of "penetrate and patch". Typical applications software and "best commercial practice" protection mechanisms will always have exploitable flaws. Patching the known flaws is necessary to deter opportunistic

attackers. Auditable designs that are free of flaws are necessary to deter motivated professionals.

8) Discretionary security policies and application policies generally are difficult to enforce in the face of malicious software, but they are effective for enforcing policies based on the user's identity or the user's role as in *role based access controls* (Ferraiolo, 1992).

9) System security engineering should ensure that, for access control policies, enforcement of mandatory policies take precedence over discretionary policies, i.e., that discretionary policy enforcement mechanisms should be unable to override the mandatory policy, regardless of discretionary permissions granted

10) Additional security requirements related to the accountability of individuals for their security-relevant actions in the computer system are required for the correct enforcement of mandatory and discretionary policies. These include such things as *identification* and *authentication*; *audit*; and *secure administration* of the system.

11) Interaction between users and protection mechanisms (e.g., authentication) requires a *trusted path* that the user can reliably invoke to ensure communication with the protection mechanism and not a malicious "spoof" of the protection mechanism. An example of such a spoof is when a user thinks that an e-mail message has been electronically signed, and yet malicious software causes an altered version of the message to be signed.

12) Encryption can provide supporting functions such as communications security, however there are limitations to the problems that can be solved by encryption. In addition, use of encryption introduces the need to manage encryption keys and reliably process the data and keys, and this often results in subtle vulnerabilities.

13) The security policy must address the potential of "indirect access", which is access that occurs outside of the perimeter of the secure computer system. This includes information that is exported from (e.g., printouts, graphic images, etc.) and imported (e.g., data, programs, software updates, etc.) to the system. For example, a system handling highly sensitive information could routinely release certain images (e.g., satellite photographs) for consumption by non-sensitive systems. Malicious software on the sensitive system could encode highly sensitive information into the image without visibly altering the image (Kurak, 1992). In response, the security policy may require assurance that released images are bit-for-bit identical to the original image acquired by the sensitive system.

Malicious Software

The lessons enumerated above include distinctions between mechanisms that are effective in the face of malicious software, and those that are not effective. A significant value of the game is its ability to illustrate the power and risks of malicious software.

Computers do as they are told. This is achieved through the execution of "programs" consisting of logical instructions. The author of the program tells the computer what to do. Attackers can author programs that execute as "malicious software". Two general forms of malicious software are Trojan horses and trap doors. A Trojan horse is typically included as part of what appears to be a "friendly" application program. Examples include viruses, worms, and logic bombs. A Trojan horse is often designed to provide the attacker with unauthorized access to information using the user's own authorizations while providing an apparently useful function. A software trap

door is typically included in programs that otherwise provide some form of access control (e.g., the underlying mechanism that determines if a given user is authorized to access specific data). By design, trap doors are often triggered by unique data (e.g., an obscure character string), resulting in unmediated access by the attacker to the system resources. Trojan horses and trap doors can be designed to be practically undetectable. Most computer systems have little or no protection against a well-motivated attacker's use of malicious software thwart security policy enforcement. Most of the "trusted" or "secure" versions of mainstream vendor operating systems are not designed to counter this type of threat. Even when these vendors have the security of their products evaluated, they do so at low levels of assurance -- well below that needed to counter malicious software. The risks associated with malicious software should be considered when selecting the minimum assurance that a given security policy is correctly enforced.

The value of the assets protected in a player's enterprise will increase as successful play proceeds. In parallel, the threat of attack by professional adversaries using carefully constructed, clandestine malicious software increases. The player must make choices with respect to network topology and security mechanisms (and their assurance) that will be effective in preventing defeat by malicious software.

Game Play Considerations

The player will have to navigate within environment shaped by two very different, yet valid worldviews:

1. While there is no such thing as "perfect security", computer systems can usually be protected by a combination of firewalls, anti-virus, intrusion detection, hardening, and monitoring.
2. Any system whose security depends on a combination of firewalls, anti-virus, intrusion detection, hardening and monitoring should not be used to protect critical resources when an adversary has a strong motive to compromise the resources.

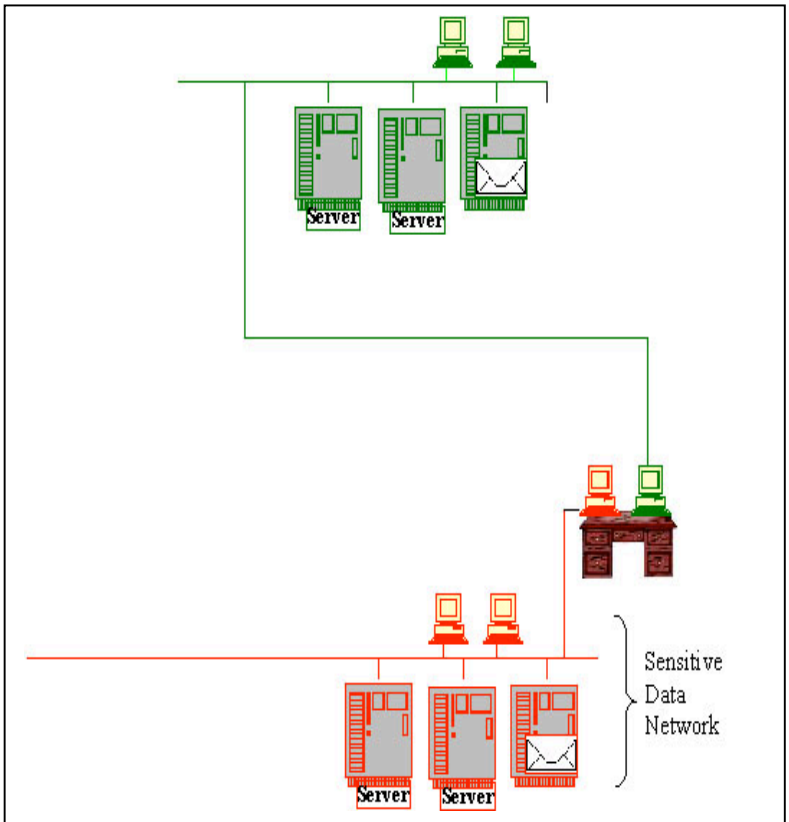

Figure 2: Physically Isolated Networks

Game Strategies

The first (and recurring) lesson the player must learn is that security policy must be understood: what resources are being protected, and from whom are they being protected? Once the player understands the value (sensitivity) of the resources (information), the player makes choices that affect the protection of the information in accordance with the security policy.

The single most important set of decisions the player makes involves the prevention of direct or indirect access to highly sensitive information by those who are not authorized to access such information. The task of the player is to construct the network so that the work factor required to attack it via technical mechanisms will exceed that possible through other avenues, e.g. social engineering or traditional espionage. Whenever presented with a valuable target and an opportunity, professional attackers (e.g., spies, industrial espionage agents, etc.) will employ tools such as malicious software to compromise the sensitive information. Separating highly sensitive information onto a physically isolated network (i.e., the "Sensitive Data Network" illustrated in Figure 2) is one solution that avoids massive losses, however it leads to operational inefficiencies and virtual user complaints due to an inability to share information across sensitivities in real time. An alternate solution that prevents massive loss while permitting controlled sharing involves the use of components that enforce a mandatory access control policy with a high degree of assurance. The challenge with this alternative is a considerable lack of availability of compliant products. A variety of other alternatives are available to the player -- each advertised as enabling controlled sharing of data in real-time while protecting highly sensitive information -- but each of these alternatives results in massive loss at the hands of suitably motivated professional attackers. Within the general scope of protecting highly sensitive information, the player can make a number of supporting policy choices related to cryptography, user authentication, physical controls and audit. Proper choices must be made in these areas in order to coherently protect the highly sensitive information, however no combination of choices regarding these supporting policies can counter wrong choices on basic access controls (e.g., the use of a low assurance product to enforce a mandatory access control policy for the protection of highly sensitive data).

Quantitatively, most of the choices made by the player have little to do with protecting highly sensitive information from people who should be constrained from accessing such information. Operational goals (e.g., an employee's ability to efficiently do work) drastically limit the scope of information that can be properly handled as if it were highly sensitive. Information whose sensitivity is moderate to low is placed on a "Non-sensitive Data Network" that provides access to great numbers of people using a range application programs. This is a natural result of people wanting to share information with the least possible interference. As a result, a good deal of information that is somewhat sensitive is placed on a network whose security depends on firewalls, intrusion detection, anti-virus and other fundamentally weaker techniques. Nevertheless, this is a viable strategy because: 1) the value of the information does not strongly motivate professional attackers; 2) compromise of the information does not result in massive loss to the enterprise; and 3) it supports easy sharing and access to information for those people who need it to do their jobs. Of course, this only succeeds if the truly highly sensitive information is kept on a physically separate Sensitive Data Network (that might be interconnected to the Non-sensitive Data Network via components having high assurance mandatory access controls). The low-to-moderately sensitive information does need protection, because all potential users are not authorized to access all of the data on these networks. Successfully protecting this information requires the player to make a lot of choices, particularly when the information is potentially accessible to large numbers of users (e.g., via direct or indirect connections to the Internet). Although, none of these choices will significantly reduce the professional attacker's ability to compromise this information, poor choices result in loss due to vandalism, disgruntled employees, incompetence and professionals willing to glean moderately valuable information from weakly

protected systems. Due to the nature of the policies and mechanisms, there are really no truly "correct" choices, so the incidence of vandalism and such is never completely eliminated.

Some of the choices the player makes towards protecting information on the Non-sensitive Data Network involve security concepts such as discretionary access controls, user authentication, audit and communications security concepts including cryptography. Other player decisions and actions may relate to ensuring that application programs are protected from unexpected input data, resulting in malicious or otherwise unintended behavior by the application itself.

At least as important as (if not more important than) the technical solutions chosen by the player are the choices related to communicating with virtual users. Understanding how the system users access information, training users, coordinating changes and being aware of what features users introduce into the system are critical to keeping virtual users happy, productive and relatively free of disaster. In addition, users must apply the procedures necessary to complement the technical security mechanisms, and this requires both user awareness and training.

Once the basics of the game are mastered, game-play converges on distinctions between "highly sensitive" information and "moderately sensitive" information. This is where the most costly losses can occur. Moving moderate-to-highly sensitive information to the Sensitive Data Network results in less information compromised by professional attackers. However, if there are limited means of sharing data in real-time across the networks, user productivity suffers. On the other hand, leaving borderline highly sensitive information on the Non-sensitive Data Network and improving its protections beyond some basic level will consume considerable resources (the player's budget) and only achieve modest gains in real security. A significant lesson in this part of the game is that attempts to create a "protected intranet" for "moderately sensitive" information through the use of firewalls will fail to protect the information from professional attackers who employ malicious software. On the other hand, actually creating a third, "Semi-Sensitive Data Network" can succeed if the interconnections with other networks adhere to the security policy and are enforced with a high degree of assurance. But again, such choices can adversely effect the ability to share data in real-time.

Protecting data within the Sensitive Data Network from people who are authorized to access the Sensitive data network, but not necessarily all of the information contained therein, is a different facet of the game. Here, authentication, audit and discretionary access control mechanisms are used to enforce a discretionary access control policy. Also, there are potentially well-motivated malicious insiders who try to amass a large body of the highly sensitive information with seriously hostile intent (e.g., to sell to competitors). Most protection mechanisms deployed to thwart this activity are defeated if the motive is strong enough. Here the only successful solution may be the recognition of an even more granular mandatory policy, which will require additional enforcement mechanisms in the form of additional separate networks and/or components providing high assurance mandatory access controls.

The most substantial variation on the game is the introduction of mandatory data integrity policies. Initially in the game, the meaning of "highly sensitive" relates to the confidentiality of the information, i.e., the information is valuable because it is a secret such as a proprietary manufacturing process. Within a single network (e.g., the Non-sensitive Data Network) integrity issues arise in the context of a discretionary policy (e.g., someone mistakenly, or deliberately, alters someone else's data). Enforcing a mandatory integrity policy introduces new challenges. While it remains the case that much can be achieved by separating the highly sensitive informa-

tion onto a distinct Sensitive Data Network, the set of people with potential access to the information now grows to include each author of software and data resident on that network.

Introducing Integrity Policies

Protecting secrets within a computer system is hard when faced with malicious software. Protecting the integrity of critical systems and data is arguably very much harder. To maintain confidentiality, the system can be physically isolated with only authorized people permitted physically access to it, as is done in "system high" or "dedicated mode" processing. On such a physically isolated system, software of dubious origins can be installed and executed without fear that the system will pass secrets to unauthorized users because their are none. But, if the integrity of data is of concern, then one must think about who wrote the software you are running, and who provided all of the data that enters your system. The player can ensure that only authorized people physically access high integrity systems, but, if so designed, the application and operating system software of unknown origin can modify data independent of the desires of the authorized users. Who wrote the software for the laptop? Who wrote the software in critical military systems? Techniques that counter subversion of secrecy policies (e.g., physically separate networks) are not always sufficient for countering the subversion of integrity policies.

For a relatively modest investment, an enemy can design, implement and field software to share in the control of a variety of different critical systems ranging from aircraft carriers to water treatment plants to electrical power distribution facilities. The problem is not one of "software safety" or "software reliability". A serious threat is deliberate design choices made by adversaries to alter the behavior of the system (Anderson, 2002). Briefly consider the fundamentals of method, opportunity and motive:

- Means: writing compact, obscure, data-driven software that performs complex operations is a fairly common skill.
- Opportunity: commercial off-the-shelf software. Go to work for a vendor and implement your own features. A recent version of the Excel spreadsheet program contains a flight simulator game triggered by an obscure keystroke sequence. No amount of testing performed by the vendor or the buyer could detect a small, well crafted, and obfuscated artifice.
- Motive: we hardly need reminding that there are relatively sophisticated groups willing to invest a lot to cause massive damage.

It may be that critical systems are controlled by carefully designed software built by reliable programmers. But in many cases this high integrity software shares the network with software that literally could have come from anywhere. The most egregious examples are high integrity critical programs that run on top of bloated commercial operating systems having no end of security flaws and tens of thousands of different authors employed by scores of different enterprises. Who wrote these operating systems? The game will include scenarios that illustrate the risks and countermeasures to the subversion threat to integrity policies. These risks are mitigated by use of high-assurance integrity enforcement mechanisms together with the use of high integrity applications for the input and modification of high integrity data (Irvine, 2002).

Assessing the Player's Success

During the game, the player's success is reflected in terms of how well the enterprise is doing (e.g., using a metric of dollars.) The player can query the "thoughts" of individual virtual users to view the user's level of happiness and current complaints and desires, such as: "I sure would like more convenient Internet access." After game play has completed, the player experiences a debriefing in which the player choices are critiqued by the system. To facilitate education and training, scenarios can be narrowly constructed to focus on specific topics. Lessons that should be learned in earlier scenarios can be incorporated into subsequent scenarios to gauge the player's ability to apply lessons learned to new situations.

Conclusion

CISR has concluded that an entertaining game for effectively teaching fundamentals of computer security can be built and is currently constructing such a game. This paper describes the abstract components of the resource management simulation. The game can be played in either of two modes: attack and defend. In each, the player makes choices with respect to security mechanisms and must interact with a set of virtual entities. When played in defensive mode, success is defined as profitable running of an Internet-connected enterprise, while offensive mode allows the player to understand the adverse impact of malicious activity.

We have provided a high level description of the network topologies to be presented by the game and have discussed important teaching objectives associated with the game. The game has been designed to be extensible. Its initial version will be restricted to single players. A subsequent version of the game permitting multiple players and advanced use of mobile computing resources is planned. Our intent is to construct the game such that it can be tailored to particular teaching objectives and will present interfaces that can be used to add supplementary educational materials as well as student assessment tools. Whether the significant investment in a game has resulted in a high payoff in education, training, and awareness regarding information assurance concepts will be the subject of additional research following its release.

References

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.

Bishop, M. (2003). Computer Security, Boston, MA: Addison-Wesley.

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

GAO. (2002). *Computer Security Progress Made, But Critical Federal Operations and Assets Remain at Risk.* Retrieved November 21, 2002 from the World Wide Web http://www.gao.gov/new.items/d02490t.pdf

Ferraiolo, D., and Kuhn, R. (1992, October). Role-Based Access Controls, *Proceedings of the 15th National Computer Security Conference*, Baltimore, MD: National Institute of Standards and Technology, 554-563.

Ham, T. (2000, March 15). *Take a walk on the virtual side of life*. Retrieved November 21, 2002 from World Wide Web http://www.usatoday.com/life/cyber/tech/review/games/cgg152.htm

Irvine, C. and Levin, T.(2001, July). Teaching Security Engineering Principles. *Proceedings Second World Conference on Information Security Education*, Perth, Australia, 113-127.

Irvine, C. E. and Levin, T.(2002). A Cautionary Note Regarding the Data Integrity Capacity of Certain Secure Systems. *Proceedings of the Working Conference on Integrity and Internal Control,* ed. M. Gertz, E. Guldentops, L. Strous. Norwell, MA: Kluwer Academic Publishers, 3-25.

Irvine, C. E., Warren, D.F., and Clark, P.C. (1997, October). The NPS CISR Graduate Program in INFOSEC: Six Years of Experience. *Proceedings of the 20th National Information Systems Security Conference*, Baltimore, MD: National Institute of Standards and Technology, 22-30,.

Kahn, D. (1967). *The Codebreakers*, New York, NY: Macmillian.

Kirriemuir, J. (2002). *Video Gaming, Education and Digital Learning Technologies*. D-Lib Magazine. Retrieved November 27, 2002 from World Wide Web http://www.dlib.org/dlib/february02/kirriemuir/02kirriemuir.html

Kurak, C. and McHugh, J. (1992). A Cautionary Note on Image Downgrading. *Proceedings of the Eight Annual Computer Security Applications Conference*. IEEE Computer Society Press.

Pfleeger, C. (2003). *Security in Computing*, 3rd Edition. Englewood Cliffs, NJ: Prentice Hall.

Rieber, L. P. (1996). Seriously considering play: Designing interactive learning environments based on the blending of microworlds, simulations, and games. *Educational Technology Research & Development*, 43-58.

Sterne, D.F., (1991). On the Buzzword "Security Policy", *Proceedings of the IEEE Symposium on Reseach in Security and Privacy*, Oakland, CA, 219-230.

Summers, R. (1997). *Secure Computing: Threats and Safeguards*. New York, NY: McGraw Hill.

Winterbotham, F.W. (1974). *The Ultra Secret*. New York NY: Harper and Row

Biography

Cynthia E. Irvine is an Associate Professor of Computer Science at the Naval Postgraduate School (NPS) where she is Director of the Cebrowski Institute for Information Innovation and Superiority and Director of the Center for Information Systems Security Studies and Research. Dr. Irvine received her B.A. degree from Rice University, Houston, TX and her Ph.D. from Case Western Reserve University in Cleveland, OH. Professor Irvine has published over 70 papers and reports on her research in computer and network security. She received the Navy Information Assurance Award in 2001, is a senior member of the IEEE and serves on several national boards.

Michael Thompson is a Research Associated at the Naval Postgraduate School (NPS). He received his B.S. degree from Marquette University in Milwaukee, Wisconsin, and has over fifteen years experience in design and development of high assurance computer and network systems.