Modeling Network Attacks

Major Scott D. Lathrop[1], Lieutenant Colonel John M. D. Hill[2], and Lieutenant Colonel John R. Surdu[1]
[1] Information Technology and Operations Center
[2] Department of Electrical Engineering and Computer Science
United States Military Academy
West Point, NY 10996, USA
(845) 938-2056
scott.lathrop@usma.edu, john.hill@usma.edu, john.surdu@usma.edu

ABSTRACT
There is rarely enough time and resources for students to explore network construction, service provision, demand analysis, and information assurance issues to the depth that they merit. What is required is a software tool that provides just such exploration with a simple interface, a supporting simulation, rapid feedback, and the ability to add ever-more sophisticated models of services / demands and attacks / defenses as the students' understanding increases. If such a software tool were available, making it web-deliverable would enable a new level of outreach to potential students of networking and information assurance. A significant aspect of the project focuses on modeling the behavior of cyber attackers and the defensive behavior of the technical and non-technical countermeasures employed by the user.

KEY WORDS
Cyber Attack Modeling, Simulation Tools, Information Assurance

1. Introduction

Teaching people how to design, build, and evaluate networks, whether for education and training or for industry, usually requires a significant investment in time, equipment, and other resources. Including the consideration of services and demands placed on that network and information assurance concerns makes it even more complex and resource intensive. There is no practical way students can take the laboratory hardware and software with them to practice network configuration and evaluation on their own. All of this places limits on reaching out to potential students who might desire to learn something in the areas of networking and information assurance. A software tool that allows students and practitioners to virtually construct networks; install services; harden their systems using security measures; place demands on their configuration; and evaluate the results, for simple to very complex setups, is sorely required for education and training. Once such a tool is available, the maximum benefit can be gained from it by making it available across the web.

2. Related Work

Several simulations exist that model either networking concepts at the byte level or strategic level Information Assurance principles. Few simulations focus on the tactical employment of information systems in a computer network. Furthermore, a few researchers have separately developed models of cyber attacks and behaviors of such attackers for the purpose of planning a defense and risk assessment. We know of no one that has attempted to incorporate such models into a simulation for the purpose of education and/or training.

OpNet Modeler, a "state of the art modeling and simulation environment that accelerates R&D for engineers designing network equipment, communication protocols, and systems" [1] and similar very sophisticated simulations are available for the examination of network hardware configurations, the transmission of packets, and the use of protocols. They are useful for student instruction, but have a fairly large learning curve, are too detailed for the novice student, and do not address the range of issues that arise in information assurance or information security instruction.

The CyberProtect application is an "interactive computer network defensive exercise … intended to familiarize players with information systems security terminology, concepts, and policy." [2] It portrays a mix between strategic-level information warfare issues to low-level network security concerns. CyberProtect takes into account some "soft" evaluation metrics such as purchasing of computer hardware and software, computer security tools, and training.

SimSecurity is a project under development at the Naval Postgraduate School that will "create a distance learning information assurance [IA] lab … packaged as an interactive game" in which the "player may perform various roles involved in IA." [3] This product is based on the idea that if city management can be made enjoyable in SimCity™ (a registered trademark of Electronic Arts), then information

assurance can be made enjoyable in an interactive gaming experience, too. It is an interesting approach to raising information assurance awareness, includes attack and defense modeling, and is designed for web-deliverable scenarios. However, it may have too little detail on network construction and evaluation and too little emphasis on the installation and management of services.

Several individuals have attempted to classify cyber attackers into a taxonomy based upon their skill and motivation for the purposes of focusing their risk assessment process. [4] Figure 2.1 is an example of such a taxonomy. A typical script kiddy could be classified as an *enthusiastic explorer* or an *enthusiastic delinquent* because they are a user of exploits but are "hacking" primarily for curiosity and/or to cause minor harm to a system for bragging rights amongst their hacking buddies. A high probability exists that such an individual possesses known reconnaissance and exploitation tools that signature-based intrusion detection systems would identify and a properly configured firewall would block. These types of threats are documented by numerous researchers and expose primarily known vulnerabilities in un-patched systems. Such individuals seek soft targets and stay away from more secure sites. Home Internet users, educational institutions, businesses, and government type organizations are primary targets of these attackers. They are the subjects of much of the work done in the Honey Net Project. [5]

Figure 2.1: Adversarial Threat Taxonomy

At the opposite extreme in the taxonomy is the *maestro cyber warrior*. This type of threat is from an individual, state-sponsored, or sponsored non-governmental (*NGO*) terrorist organization. *Maestros* are very skilled programmers and have a deep understanding of operating systems and network protocols. Their exploits attack unknown and/or unpublished vulnerabilities in information systems. Targets may include government, military, and Supervisory Control and Data Acquisition (SCADA) networks.

A few process models have been proposed to model the types of attack that an information system may be subject to. Attack trees and petri nets are the most favorable representation. [6, 7] The models are used primarily for focusing efforts and resources during risk assessments or penetration testing of an information system. Modeling attacks is primarily used to "war game" the possible threats that an information system may be subjugated. Again, we know of no one who has made an effort to incorporate such representations in a simulation.

Attack trees provide a methodical way of describing system security based on the types of attack. [6] Figure 2.2 shows an example attack tree. The root node (G_0) represents a goal the attacker is attempting to achieve. Each node in the graph represents a set of sub-goals that must be achieved in order for the top-level goal to succeed. Sub-goals may be represented as an AND-decomposition or an OR-decomposition. In order for a goal with an AND-decomposition to be achieved, all the sub-goals must succeed. A goal with an OR-decomposition represents a choice, where at least one of the sub goals must be achieved in order for that goal to succeed. The leaf nodes in the attack tree represent the instantiation of an actual attack. Given the attack tree in Figure 2.2, there are two possible successful attack instantiations: $\{G_3, G_5, G_6\}$ and $\{G_4, G_5, G_6\}$. The use of OR-decomposition result in new instantiations of attacks while adding a node to an AND-decomposition extends the requirements for an existing attack. [8]

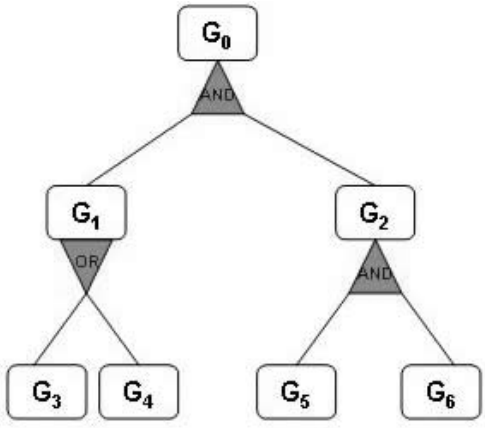


Figure 2.2: Example Attack Tree

3. MAADNET Project

The Military Academy Attack/Defense Network (MAADNET) is a multi-module system that will provide network construction, evaluation, attack, and defense capabilities for the classroom and beyond. The purpose of the MAADNET project is to addresses the problems of determining where risk in the system is acceptable and then making resource and employment decisions that minimize risk where compromise is unacceptable. The designers are working towards a product that compares individual designs against each other rather than against a system "solution". The focus is on the technical and tactical details of how to employ a secure information system to include soft factors that further enhance or degrade the overall system. Such soft factors include system administrators' qualifications, user training, and policies in place. [9]

The MAADNET project is broken down into several modules. They include network construction and evaluation, service and demand modeling, traffic modeling and simulation, attack modeling, and web-delivery and competition.

The contributions of the MAADNET project include an easier to use network construction interface, a service / demand focus to network modeling, a less-detail oriented traffic model, the ability to realistically portray attacks over time, and a new way to reach out to potential students. The results of this research should be useful to others interested in networking and information assurance, particularly for training and education.

The network construction module includes a visualization of results and an eye to the long-term goal of web-based delivery and competition. The interface takes advantage of modern drag-and-drop and visualization techniques to enable rapid construction and evaluation. Once the hardware of the network has been set up and the links established the user must configure the services provided over the network in support of demands specified in a given scenario. A discrete-event simulation mechanism will model traffic generation and flow through devices and across links.

Once the network is built and services established the user will submit their design through a web-based delivery mechanism where it will be subjected to a series of attacks based on the scenario. The system will evaluate the performance of the design by measuring its ability to maintain confidentiality, integrity, and availability of its information against the attacks as described by [10].

Use of MAADNET is scenario driven. The scenario might indicate, for instance, that in the user's organization confidentiality of information is vitally important. A successful attack against availability, then, would have less impact on the user's evaluation than would a successful attack against confidentiality.

The attack subsystem of MAADNET consists of several attack agents modeling different types of threats. These attacks will specifically target the confidentiality, integrity, and availability of the defended information system using several possible, non-scripted approaches. The probability of an attack succeeding is based on the type of attack, the attacker's skill and motivation, the defense emplaced by the user, the skill of the system administrator(s), and amount of user-level training invested in by the student. Both technical and non-technical attributes are important. Initially these attack probabilities are determined in a subjective manner, through interviews with experts and research. Developing solid models of these attack-defense relationships remains an open research issue.

4. Modeling Cyber Attacks

Defenses are static in the simulation. That is, after the user has configured their defensive plan, the defensive mechanisms will not change. However, the attackers' behavior is dynamic based on the scenario and so their activities must be modeled. This section describes how such cyber attacks are modeled.

Since the defense is static, modeling the defensive behavior of the information system involves determining if a counter measure exists at any node in the attack tree as an attack agent proceeds through their possible contingency plans. In MAADNET a set of finite choices for each possible security measurement will be provided to the user. The security countermeasures may be in the form of technical solutions, policy and procedure, and/or the qualification, education, and training of the systems administrators and users. The user must decide what tools to buy or gather if they are open source and where in the information system to employ those tools. Tactics such as defense-in-depth, aggressive vulnerability scanning, annual training of users and system administrators, and establishing and enforcing password policies all reduce risk and the probability of an attack agent succeeding. Since the user will have limited resources in terms of money and time, they will have to take into account the type of organization they are manning in order to determine where risk is acceptable. Soft factors (i.e. policy, procedure, administrator qualifications, etc.) will adjust the probability at each node in the attack tree also.

An attack agent in MAADNET may begin an attack from one of three locations (Figure 4.1). The determination of where an attack may begin is decided by the educator or

trainer of the student. The first location an attack(s) may commence from is the Internet. Such an attack starts from an external location. In its start state, the attack agent does not have access to user accounts in the defended network. The second starting point for an attack agent is from an internal node on the network. Such an attacker is considered an "insider" as they have, at a minimum, a user account on a local machine, or worse a user account in the network domain. Finally, an attack may begin from a position that is within geographical proximity to a wireless segment of the defender's network. Such an attack could be from either an external or internal (i.e. "insider") threat.

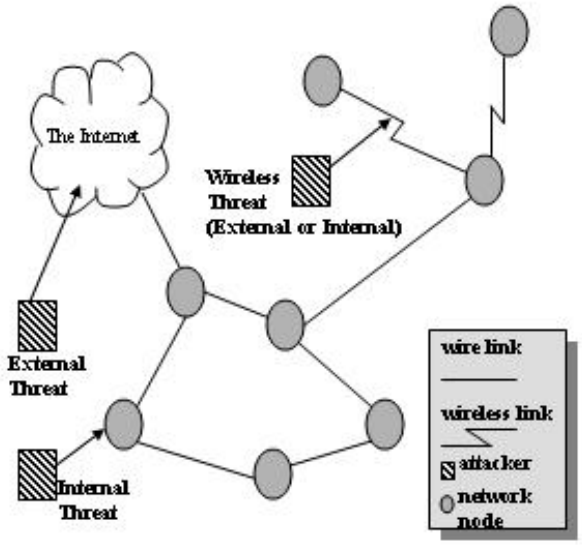


Figure 4.1: Possible Attack Agent Start Points

An attacker's behavior is modeled in two ways. The first is with a modified version of attack trees that represent both the skill and motivation of the attacker. The second is through the decision-making process by which branches of the attack tree are chosen. The attacker's behavior must take into account the current phase of their attack, the given scenario, and the attacker's particular goal(s).

We define the phases of an attack to be reconnaissance, exploitation, and consolidation and reorganization. Reconnaissance includes determining key information that allows an attacker to successfully execute a particular exploit. Key information includes Internet protocol (IP) addresses, open ports, types of operating systems and applications running on the end system, and firewall rules. Exploitation includes the actual attack on the system to include buffer overflow attacks, viruses and worms, and password crackers. We define consolidation and reorganization to include those tasks an attacker may carry out to hide their activity and keep control of the victim platform or network. These may include backdoors, root kits that erase logs or replace commonly used system commands, and encryption techniques to secure their transmissions from eavesdropping.

The educator or trainer initially chooses what tools the attackers have available to carry out reconnaissance, exploitation, and consolidation. The probability of the appropriate tool being chosen is based upon the type of attacker being modeled. In creating the scenario the educator or trainer describes in generalities the risks the network will face. For instance, the scenario might read:

"As a bank, confidentiality and integrity are much more important to you than availability. Your bank is small, but banks of your size are frequent targets of "script kiddies." Four or five times a month you can expect an attack from a criminal, and you will infrequently be attacked by a cyber warrior. Your network consists of both wired and wireless LAN segments in order to support mobile computing."

Given the scenario described, we have defined three classes of attackers: low-skilled, explorers or delinquents which we call *script kiddies*. They are inexperienced hackers who are generally capable of running attack scripts someone else has written. They usually do not understand the code or even the particular types of services the exploits work against. For example, they may launch a web exploit against any machine running a web server, even if that machine is not running the particular web server for which the exploit was designed.

The second class of attacker we define is a *criminal* who has illegal or unethical motivation and moderate programming skills to either create their own simple exploits or modify existing exploits. Finally, our elite hacker is a *cyber warrior* who has possible state sponsored or terrorist support and skills to develop very sophisticated attacks for both known and unknown vulnerabilities against various types of operating systems and applications.

Each agent has access to two primary data structures: an attack tree and a state table, or working memory that accounts for the attacker's current view of the world. An abstract attack tree is shown in Figure 4.2. Our version of the attack tree has subtle differences from the attack trees described previously. We use both AND-decompositions and OR-decompositions. Additionally, we create a COND-decomposition (conditional decomposition) and define preconditions and post conditions for each node in the tree (for simplicity only one pre/post condition example is shown in figure 4.2).

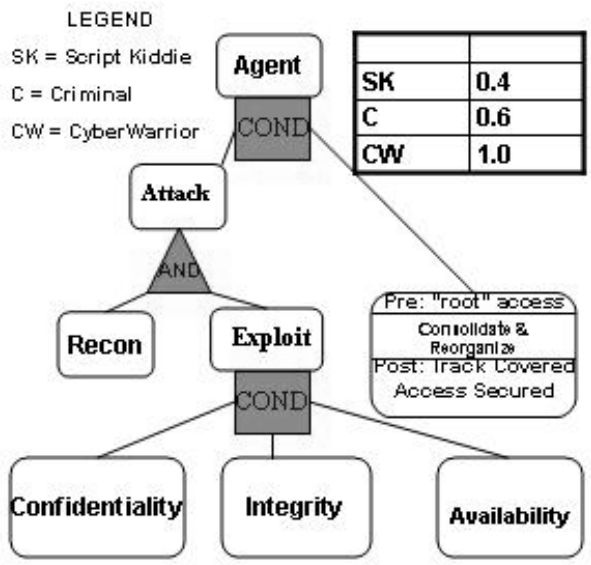


Figure 4.2: Abstract Attack Tree

COND nodes indicate that an agent may decide whether or not they want to achieve the goal. For the agent to traverse a COND node, two questions must be answered by the agent: (1) do I want to perform this action, and (2) are the necessary preconditions met for me to take this action? The answer to question one is determined by a probability table based on the type of attacker. The answer to the second question is satisfied by a lookup to the agent's state table. If the necessary preconditions are met then the agent can continue down the branch in the attack tree. Otherwise, the agent has to readjust their position in the tree to the first node where the desired precondition is satisfied. They then continue their attack down that branch of the tree.

For example, Figure 4.2 indicates that in order for an attacker with a goal of compromising the confidentiality of information within the defended system, the agent will have to successfully perform reconnaissance and successfully execute an attack that compromises the confidentiality of the system. They may or may not choose to consolidate on certain objectives that they have achieved. If they decide to consolidate such as by installing a back door or covering their tracks, certain preconditions will have to be met during the reconnaissance and exploitation phases of their attack. If those conditions are not met, they may decide to return to specific portions of the reconnaissance or exploitation branches in order to satisfy those conditions. Otherwise they may choose to do nothing. The latter are actions that a typical script kiddy may take, while a criminal or cyber warrior would more than likely want to consolidate on their objective.

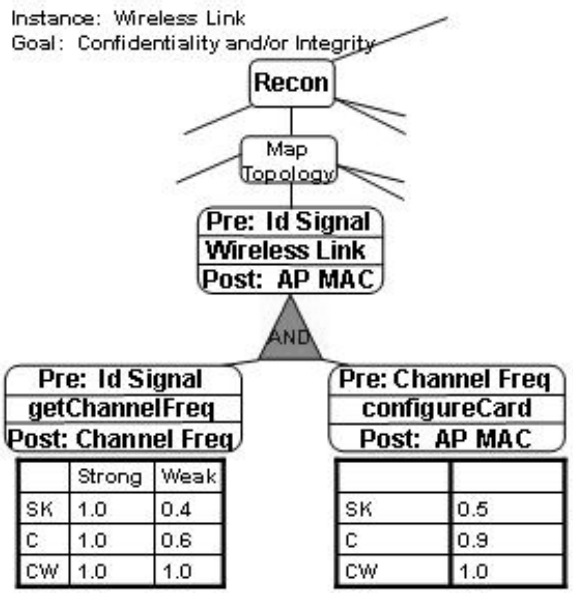
Such behavior is indicated by the probability table associated with the branch from the *Agent* node to the *Consolidate and Reorganize* node. In this case a script kiddy would attempt to consolidate with a probability of 0.4 while a cyber warrior would definitely consolidate on their objective. If the decision is made to consolidate then the agent has to make sure they have met the necessary preconditions for that branch. In Figure 4.2 this is indicated by a precondition of root/administrator access gained on a host machine. If the agent did not have such access because perhaps their exploit allowed them to only gain user level access, the agent would have to backtrack in the tree to a point where they could continue their attack. In this particular case the agent could continue their attack at a point where access to a particular host machine was a precondition and root level access was a post condition.

When traversing through a sub tree of AND nodes the policy is to traverse the tree from left to right. This allows the developer of the attack tree to specify ordering of actions that must be performed in sequence. Currently the methodology does not allow for two events that must occur simultaneously. For OR nodes the choice is important and this is determined by the behavior of the attacker. Again, this behavior is initially going to be modeled using probability tables.

The state table, or working memory, accounts for what conditions the attack agent has satisfied. This provides the means for determining whether an agent has met the necessary preconditions for either sequential steps in a current attack or branches to new attacks. Continuing with our previous attack, if an agent has successfully gained user level access on a particular platform, it may want to continue to exploit that machine by attempting to gain administrator level access. Additionally, the agent may want to branch to a new attack sequence by starting reconnaissance from the machine in order to gather further information. In the simulation this would be implemented by the spawning of a new attack agent with the attack type and state table of the parent agent.

As a very specific example, we will look at an attack agent that begins its attack as a threat to the wireless network. Such an agent would have its state table initialized with the necessary equipment to detect a wireless signal (i.e. wireless card, antenna, and software). The reconnaissance phase of such an agent may be modeled as in Figure 4.3. Goals of reconnaissance would be to identify wireless access points (AP) by their media access control (MAC). In order to accomplish this goal the attacker would have to both determine the channel the signal was being transmitted on

and have the skill to configure their wireless card to that channel.



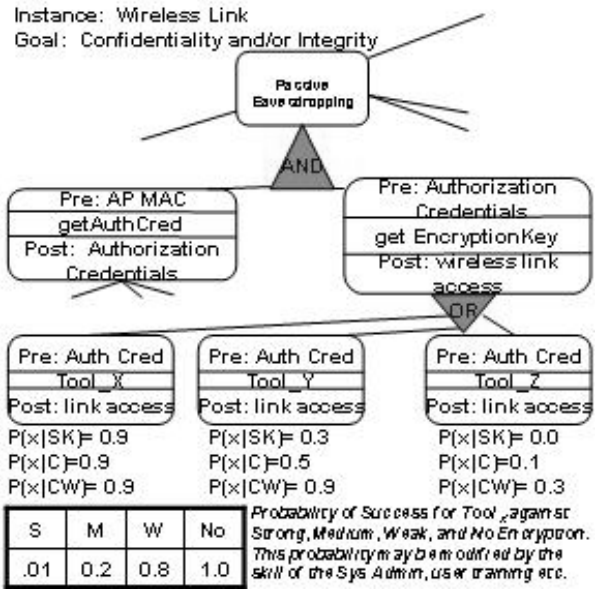
	Strong	Weak
SK	1.0	0.4
C	1.0	0.6
CW	1.0	1.0

SK	0.5
C	0.9
CW	1.0

Figure 4.3: Wireless Link Reconnaissance

Figure 4.3 highlights specific examples of our modified attack trees. First, the appropriate preconditions and post conditions at each node are indicated. Second the figure shows different types of probability tables that can be associated with nodes. The table on the left indicates the probability of the attacker being able to detect the wireless signal and gain the channel frequency of an access point's signal. The trainer can "position" the wireless attack agent at different intervals from the access points. The *strong* column indicates the attacker being able to detect a strong signal from the access point while the *weak* column indicates the attacker is detecting a weak signal. If receiving a weak signal, a skilled attacker would have a higher probability of having the knowledge to build a wireless antenna that has the capability of receiving a strong enough signal to continue the attack while a script kiddy would have a lower probability of performing the same task.

Continuing with the wireless attack, the agent reaches a point in its attack tree where it is attempting to passively eavesdrop on the communication link in order to violate the confidentiality and/or integrity of the information transmitted over that link (Figure 4.4). In this situation the agent has to both authenticate to the access point and get the encryption key. In order to get the encryption key the agent has a choice between three different tools (X, Y, and Z). The probability that an agent has access to such a tool is conditional based on the type of attack agent that is instantiated (script kiddy, criminal, or cyber warrior). Since any tool used in the situation will decrypt the encryption key, a cyber warrior has a greater opportunity of success. If Tool_X does not work, the agent can try Tool_Y as the preconditions for using Tool_Y are still in place.



S	M	W	No
.01	0.2	0.8	1.0

Probability of Success for Tool_x against Strong, Medium, Weak, and No Encryption. This probability may be modified by the skill of the Sys Admin, user training etc.

Figure 4.4: Wireless Link Exploitation

As an example of how the attack and defense behaviors of the simulation work together, the bottom table in Figure 4.4 displays the probability of success of Tool_X. This probability takes into account the quality of the exploitation tool along with the "soft factors" in the simulation related to the defense of the network. A particular attack tool may start out with a probability of 1.0 and decrease as the student hires a skilled system administrator who correctly configures the access point with encryption. If the student purchases a wireless security solution, the probability of the attack tool working decreases even more.

5. Simulation Incorporation

This section briefly describes how the attack model is incorporated into the simulation. Before the simulation is initialized, agents representing the three types of attackers are attached to the simulation as "participants." When the simulation is initialized, the simulation executive tells each participant to schedule its first event. Probability distributions are used to control the inter-arrival times of attack events from each type of attacker. For instance, the script kiddy might use an exponential distribution with lambda of fifteen minutes, while the cyber warrior might use an exponential distribution with a lambda of fifteen days. As the simulation executes, events are pulled off the event

queue in time-stamp order. When an attack event is pulled off the queue, the simulation executive calls the attack agent. Each attack agent maintains a pointer to a node in one of its attack trees. When called upon by the simulation executive to execute an attack, the attack agent finds its pointer in the tree and calls the Attack Resolution Module (ARM) to resolve the attack. The agent's pointer always points to a leaf node when the attack event is pulled off the queue.

The ARM determines the results of the agent's actions. The agent must tell the ARM which network node is being attacked and what type of attack (tool) it is executing. The ARM determines the probability of the attack succeeding based on the probability tables associated with the attack tree at that node (a real number between 0 and 1) and generates a uniform pseudorandom number. If this "die roll" is less than the probability of success, the result of the successful attack is returned to the attack agent. The pointer in the attack tree is updated, the agent updates its state table with information returned from the attack, and the attack agent schedules its next event with the simulation executive (5.1).

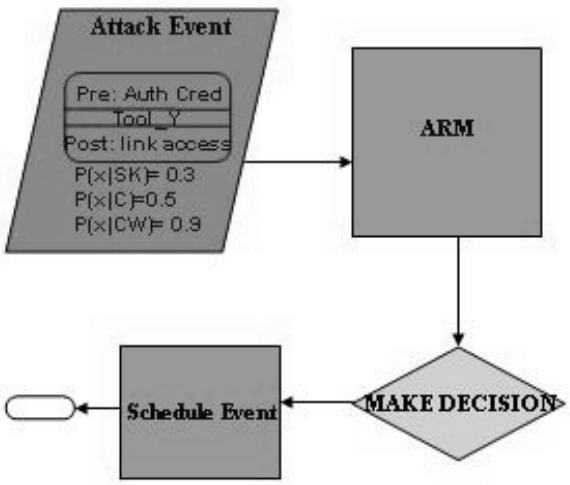


Figure 5.1: Attack Event

How does the agent update its pointer? The pointer continues to move up the tree (from child to parent) as long as the node is satisfied. Once the pointer reaches a node that is not satisfied, the attack agent begins to choose branches to child nodes until a leaf node is reached. At this point the agent has made a decision as to what attack to execute during its next attack event. This new attack event is then scheduled for a time in the future based on the duration of the attack represented by that leaf node. The duration of an attack is also defined by a probability distribution, and that duration is used to determine the time stamp of the next event.

If the returned values from the ARM sets preconditions for other attacks, the agent may decide to spawn a new attack agent with the agent type and state table of the parent agent. This decision is made by the attacking agent based upon the type of attack agent.

6. Future Work

With an initial prototype that models the network construction and evaluation foundation, the next step is to incorporate attack modeling as described in this paper. Additionally the development of a web-based delivery mechanism must be implemented for the competition.

The scenario tools file include XML descriptions of capabilities such as firewalls, intrusion detection systems, proxy servers, etc. that are available to the users of the simulation. It also includes a description of the "soft" factors associated with building an information system that is the type of people that are hired, the policies and procedures in place, and the amount of training available. The result of the user's design is another XML file that describes their implementation. This XML file will be served to the web server for evaluation against the attack models.

Concerning attack modeling, implementation of the design discussed in this paper is the first step. We would also like to be able to model the coordination between different attack agents whose attacks start at different points in the network. Finally, after successful implementation, we would like to model a dynamic defense where network nodes and links change as the user discovers flaws and modifies their design.

7. Conclusion

The MAADNET system will provide several significant advantages over traditional networking and information assurance instruction: rapid network construction and evaluation, in-class explanation by instructors, out-of-class exploration by students, the ability to develop scenarios emphasizing particular topics, and the ability to reach a large audience.

A key component of the system will be the modeling of how a cyber attacker behaves. Initially the modeling will be based on modified attack trees and the implementation will be simple table lookups to determine the results. Ultimately, we plan to migrate the attack tree modeling concept into a cognitive, agent-based system in order to provide a better evaluation than simple probability table lookups.

References

[1] OPNET Technologies, "MIL3 - Third Millenium Technologies, OPNET," http://www.mil3.com/, accessed on February, 2002.

[2] J. H. Saunders, "Simulation Approaches in Information Security Education," presented at 6th National Colloquium for Information System Security Education, Redmond, WA, 2002.

[3] M. VanPutte, "SimSecurity - Distance Learning and Virtual Laboratory for Information Assurance," http://www.movesinstitute.org/OpenHouse2001/Presentations/VanPutteSimSecurity.ppt, accessed on 19 December, 2002.

[4] D. Welch, "Adversary Threat Taxonomy," presented at IEEE Information Assurance Workshop, West Point, NY, 2002.

[5] The Honeynet Project, *Know Your Enemy Revealing the Security Tools, Tactics, and Motives of the Blackhat Community*. Boston: Addison-Wesley, 2002.

[6] B. Schneier, "Attack Trees: Modeling Security Threats," *Dr. Dobb's Journal*, pp. 5, 2000.

[7] J. P. McDerrmott, "Attack Net Penetration Testing," presented at The 2000 New Security Paradigms Workshop, Ballycotton, County Cork, Ireland, 2000.

[8] A. P. Moore, R. J. Ellison, and R. C. Linger, "Attack Modeling for Information Security and Survivability," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, Final Technical CMU/SEI-2001-TN-001, March 2001.

[9] C. A. J. Carver, J. R. Surdu, J. M. D. Hill, D. J. Ragsdale , S. D. Lathrop, and T. Presby, "Military Academy Attack/Defense Network," presented at 3rd Annual IEEE Information Assurance Workshop, West Point, NY, 2002.

[10] W. V. Maconachy, C. D. Schou, D. Ragsdale, and D. Welch, "A Model for Information Assurance: An Integrated Approach," presented at 2001 IEEE Information Assurance Workshop, West Point, NY, 2001.

Author Biographies

MAJOR SCOTT D. LATHROP is a Senior Research Scientist with the Information Technology Operations Center and an Assistant Professor in the Department of Electrical Engineering and Computer Science, at the United States Military Academy, West Point, New York. His academic interests lie in information assurance. He is the head coach of the cadets' inter-Academy Cyber Defense Exercise team. Major Lathrop is an Armor officer.

LIEUTENANT COLONEL JOHN M. D. HILL is an Assistant Professor in the Department of Electrical Engineering and Computer Science, at the United States Military Academy, West Point, New York. His academic interests lie in the application of emerging computer science technologies to automated planning and decision support systems, information assurance, tactical information flow, and visualization. Lieutenant Colonel Hill is an Armor officer.

LIEUTENANT COLONEL JOHN R. SURDU is a Senior Research Scientist with the Information Technology Operations Center and an Assistant Professor in the Department of Electrical Engineering and Computer Science, at the United States Military Academy, West Point, New York. His academic interests lie in simulation technologies and novel uses of artificial intelligence techniques for decision and operations support. Lieutenant Colonel Surdu is an Infantry Officer.