Military Academy Attack/Defense Network Simulation

John R. Surdu, John M.D. Hill, Ronald Dodge, Scott Lathrop, and Curt A. Carver
Department of Electrical Engineering and Computer Science
United States Military Academy
West Point, NY 10996
{john.surdu | john.hill | ronald.dodge | scott.lathrop | curt.carver} @usma.edu

Keywords: Discrete event simulation, information assurance, education.

Abstract

One can argue that Information Assurance education is vitally important. It is often impractical to allow students to experiment with real networks. A simulation-based tool is needed to supplement classroom instruction. This paper describes the architecture of the Military Academy Attack/Defense Network (MAADNET) that allows users to explore interrelationships between people, procedures, hardware, software, and data and how each of these factors impact on network design and security. Because we designed MAADNET to be used for education, MAADNET allows the instructor to tailor the simulation scenario based on the lesson's training objectives. MAADNET simulates various events and grades the network based on "hard" metrics like message latency, percent down time, etc. The network is also graded on "soft" metrics like how well confidentiality, integrity, and availability were maintained during simulated attacks. This paper describes the phased development effort, current state of MAADNET, and a timeline for future development.

INTRODUCTION

Students retain more information from hands-on exercises than they do from merely reading text books and listening to lectures. While teaching information assurance, however, it is often impractical to allow students to experiment with real networks. Special-purpose Information Assurance classrooms are expensive, and they require significant resources to establish and maintain. A simulation-based tool is needed to supplement classroom instruction.

We have begun the design and implementation of the Military Adacemy Attack/Defense Network Simulation (MAADNET). This effort has a number of objectives. MAADNET will support basic computer networks instruction. It will be used to support information assurance education to prepare students to enter our special-purpose information assurance laboratory. Finally MAADNET will be used as the infrastructure for a web-delivered information assurance competition

This paper describes the architecture and design of MAADNET that allows users to explore interrelationships between people, procedures, hardware, software, and data and how each of these factors impact on network design and security. Because we designed MAADNET to be used for education, MAADNET allows the instructor to tailor the simulation scenario based on the lesson's training objectives. MAADNET uses a client-server architecture in which the user builds and tests a network design on the client side and later submits the planned network to the server. The server simulates various events and grades the network based on "hard" metrics like message latency, percent down time, etc. The network is also graded on "soft" metrics like how well confidentiality, integrity, and availability were maintained during simulated attacks.

Of particular interest is the manner in which we will simulate attacks on the students' networks. We propose the use of attack trees for this purpose, as attack trees have proven to be a useful technique for modeling real attackers. This paper describes the phased development effort, current state of MAADNET, and a timeline for future development.

INTENDED USES OF MAADNET

Two primary factors that hamper successful information assurance training at a majority of education centers, whether it is a junior high school or community college, are a lack of hands-on training and a lack of tools designed to teach information assurance. It is impracticable to support even a small number of students with the physical resources required to experience hands-on information assurance training. MAADNET presents a simulation environment that provides a student with all the components needed to construct and manage an information system, addressing both the lack of hands-on training and the lack of tools to teach information assurance. The "hands-on" approach to learning used by MAADNET, coupled with its scoring module and Web-

based delivery, provides a user with an integrated, engaging, and challenging information assurance learning environment.

MAADET can be used to cover a wide spectrum of information assurance education. MAADNET is built to allow an instructor to scale the complexity of the scenario to fit the level of the audience. For example, the type of network services or hardware to be used by the student can be specified in the scenario to guide the construction of the network. Similarly, the network can be presented partially built, so that the student can concentrate on only that are the instructor thinks is most interesting. Another example of scaling the complexity includes constructing the attack tree used to attack the network to either present the user with complex attacks or with basic attacks that can be defeated with rudimentary security policies.

MAADNET can be used in a variety of educational settings. MAADNET is being developed with the capability to "de-couple" the simulation components to satisfy the requirements of a given lesson objective. For example, if an instructor is teaching basic network design, the attack and/or hardware failure events can be disabled, allowing the instruction to focus on the basics of networking. The demands placed on the network can also be manipulated to meet the goals of a lesson.

The competition feature allows MAADNET to be taken out of the "classroom teaching environment" and permit students to compete to build a superior network. As with the ability to modify the scenario in the classroom to meet given lesson objectives, the scenario used for a competition can be scaled to meet the abilities of the competitors. Previous efforts in web-enabled distributed completion have proven very successful [1].

PHASES OF IMPLEMENTATION

The MAADNET development effort is broken down into three major phases to accomplish our overall objectives: support basic networking instruction, support information assurance instruction, and provide a web-delivered competition.

The objective of phase one is to build a robust network construction tool for use in and out of class. This tool, called MAADNET NetBuilder, is targeted at supporting instruction in an undergraduate networking course; however, it is designed to be simple enough for anyone to rapidly gain an understanding of networking fundamentals.

Implementation began with an intuitive graphical user interface (GUI). The next important step was developing the ability for the instructor to create and store a scenario for the students, including the requirements, the physical layout, the equipment available, and anything else they needed. It is important to note that these scenarios can be targeted at different levels of expertise in several aspects of networking capability (see **Basic Design and Implementation**). The next important part of this phase was the development of the communication model, which is a message-level aggregation of traffic over the network devices and connections.

With the communication model in place, work has progressed in modeling services and demands placed on those services. Some of the services being modeled are file servers, Email servers, database servers, and Web servers. These services are installed on computers with a simple drag-and-drop interface. Behind the scenes, though, the service places its own load on that computer and on the network. In the service/demand model, demands against the services are mostly associated with users. Again, a simple drag-and-drop operation associates a particular user with a computer. That user brings with it a set of demands on some or all of the services available.

In response to the scenario provided by an instructor, the students build a network configuration. Even at this very basic level we have the concept of evaluation (also called scoring). The scoring function could be as basic as cost, or it could have additional factors, such as making sure everything is properly connected, ensuring that sufficient bandwidth is available to support all the demands on the system, etc.

To support the communication model and the service/demand model an underlying discrete event simulation (DES) mechanism was designed and implemented. This DES mechanism allows the student to visualize communications flow across the network and assists in the evaluation (or scoring) process.

The work on phase one is practically complete, with a MAADNET NetBuilder prototype scheduled to be tested by students in the spring semester of 2003. The objective of phase two is to enhance the network construction and evaluation tool to incorporate defense and attack modeling. This enhancement is called MAADNET Competitor, and is targeted at the instruction of information assurance concepts. The idea is to allow students to take the configurations they create in MAADNET NetBuilder and incorporate defense capabilities. Defenses are typically modeled as services, but they can also be policies and procedures.

The approach taken in this phase is that the defenses have some impact on the flow of communication (inspection, blockage, etc.) and therefore must be integrated into the communication model and the service/demand model. The same is true for the modeling of attacks against the network. As such, they are controlled by the underlying simulation mechanism and are therefore amenable to observation and analysis.

Another requirement is to integrate the effects of defenses and attacks into the user interface, enabling the user to visualize those effects. The most important aspect

of this phase is to determine correct methods for modeling the interaction of various, perhaps simultaneous, attacks with multiple defense mechanisms.

The objective of phase three is to host a Web-based competition to raise interest in networking and information technology and to raise information assurance awareness. A first important step is to deliver MAADNET Competitor over the Web. The idea is to give people anywhere the ability to download the software, accept a scenario, explore network construction and evaluation, and receive immediate feedback on their configuration. Once they master the fundamentals, they can explore how to defend their configuration. Finally, they can submit their configuration back to the MAADNET server so that it can be scored based on how well it withstands a scenario of attacks. The competition aspect comes from determining the best scores for each scenario and keeping track of those who wish to compete and voluntarily provide contact information. The initial concept is to run this in a fashion similar to the USMA Bridge Design Contest, an online competition in the engineering domain [2].

BASIC DESIGN AND IMPLEMENTATION

This section describes the basic structure of MAADNET, including how scenario data files are stored, how attacks on the network will be simulated, how the simulation is configured to meet the specific training/educational purpose, and how the simulation will be Web-delivered.

Layered Design

The implementation of MAADNET follows a layered approach. This layered approach has helped guide our prototyping as well as design. The Simulation Layer contains those services and modules common to nearly all discrete event simulation systems. The Communications Layer includes the traffic generators (described later) that create messages that travel around the network. The Device and Link Layer includes network interface cards, network connections (e.g., Ethernet cables), workstations, routers, switches, hubs, servers, etc. The Service and Demand Layer includes the services that are available in MAADNET: Email servers, Web servers, File servers, Database servers, and Applications. The top layer of MAADNET, the User Interface Layer, includes the users of the student's network and their interaction with that network. In implementing this layered design, as shown in Figure 1, we used a spiral development strategy. Once the Simulation Layer was complete, each spiral included some parallel work on new features in each of the other four layers.

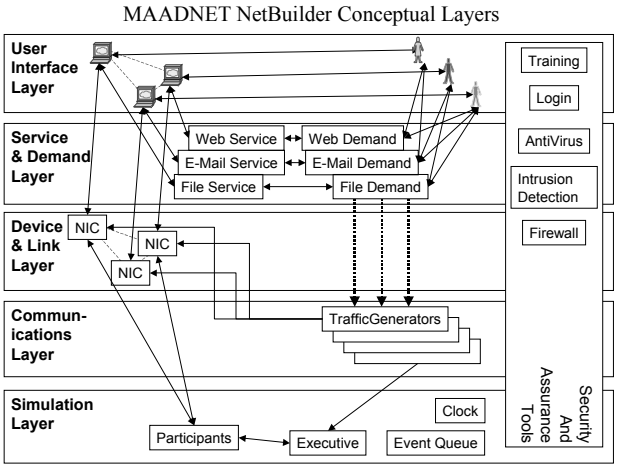

Figure 1: MAADNET Layered Design Approach

Overall Design

To make MAADNET easy for students to use, a drag-and-drop interface is used to configure the network, assign users to workstations, assign services to workstations and servers, etc. There are several scenario files associated with a MAADNET scenario. The formats of all of these files are defined by XML schema we have developed. The overall scenario file includes a textual description of the scenario for the student to read and the locations of the following files:

- Environment (i.e., the physical layout of the buildings and room(s) in which the student will place equipment and personnel),
- Scoring file (i.e., the scoring formula coefficients) (hidden from the student),
- User file (i.e., the information about each user's demands on the network), and
- Resources file (i.e., the hardware, software, system administrators, services, etc.).

The physical layout defines the "sandbox" for the simulation. The information from the other files is used by the interface to populate the simulation. For instance, the hardware (including workstations, servers, network interfaces, network cables, etc.), software (services like Email, database, file, intrusion detection systems, firewall software, etc.), and support personnel described in the resources file are used to populate a palette from which the student may draw in creating his or her network. In Figure 2 the "Component Palettes" area at the top of the GUI is populated from the user and resources files, and the "Environment" area is defined by the environment file.

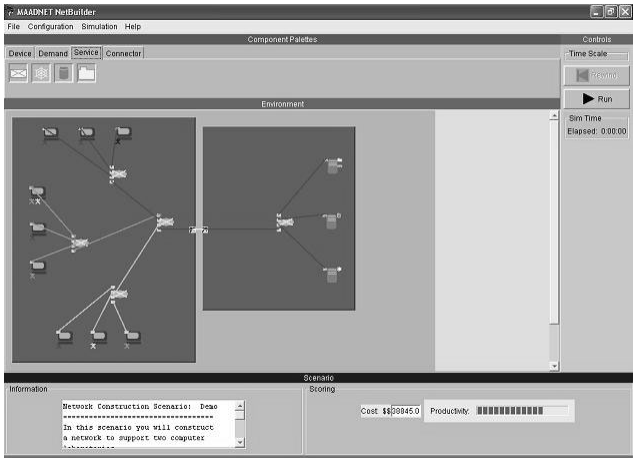

Figure 2: Early MAADNET NetBuilder prototype.

```
<user type="minion" num="2">
  <email payload="Email_message">
    <internet>
      <arrival type="normal">
      <p1>5000</p1><p2>1000</p2>
      <size    type="expo"><p1>3000</p1>
    </internet>
    <all_users>
      <arrival type="normal">
      <p1>22000</p1><p2>1000</p2>
      <size type="expo"><p1>3000</p1>
    </all_users>
    <one_user>
      <arrival type="normal">
      <p1>5000</p1><p2>1000</p2>
      <size    type="expo"><p1>3000</p1>
    </one_user>
  </email>
</user>
```

Figure 3: Example User E-Mail Demands

MAADNET is designed in accordance with a typical discrete event simulation (DES) methodology (as shown in Figure 4). Services (such as Email and Web servers), hardware devices (such as workstations, routers, and servers), and users of the student's network are not entities in MAADNET; they are referred to as Participants. Entities in MAADNET are the traffic generators associated with users of the simulated system. These traffic generators are defined in the User file. Each type of user has three usage profiles for Email (to a single recipient, to all other users in the organization, and to the Internet), a usage profile for database access, a usage profile for Web access (e.g., using a browser), a usage profile for file server access, and a usage profile for demands on the user's workstation (i.e., how much memory, CPU, and disk space the user demands). These profiles include the frequency of access as well as the size of the data being transferred. Frequency and size of messages can be constants or be defined by probability distributions. (See Figure 3 for an example of the Email profiles for a user of type "minion.")

In MAADNET *Events* are defined as the transmission and arrival of messages (aggregates of network packets, not individual data packets). Other types of events include attack events, equipment/service failure events, and equipment/service repair events. For instance, when using NetBuilder, hardware and services have a defined mean time between failures (MTBF) and mean time to repair (MTTR). When the simulation is initialized, each Participant schedules its first Event. For services and devices, this is usually a failure event, determined by that service's or device's MTBF. When a failure event is executed a repair event is scheduled. The timestamp of this repair event is based on the MTTR, but it is modified by factors like the number and quality of system support personnel "purchased" by the student.

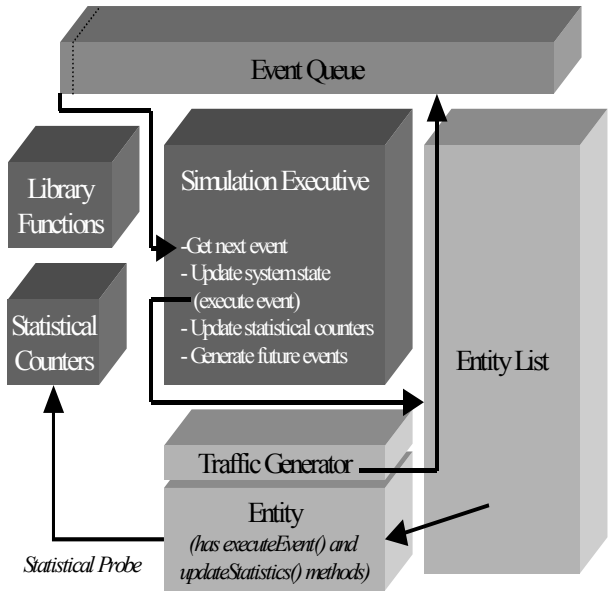

Figure 4: MAADNET is based on a typical DES design.

The interface of MAADNET uses drag-and-drop and right-click functionality to make the system easy for a student to use. For instance, the student can drag a server from the palette and drop it into the environment. Any properties of the server that can be configured by the student are available through right clicking on the icon.

Modeling Attacks and Defenses

Once services are being provided across the network and demands are being placed against those services, attacks against those services and defenses against the attacks will be incorporated. This section describes how attacks and defenses are modeled.

The modeling of attacks must take into account two considerations: (1) the threat and (2) the possible types of attacks that a attacker may employ based on the security of the system. The attack subsystem of MAADNET will consist of several attack agents modeling different types of threats. Attack agents must be attached to a hardware device in the network – just like other users. In most cases, they will be attached to the "Internet" node, representing an "outsider" attacker, but "insider" attackers may be attached to workstations within the student's network as well.

Each agent is defined by a probability distribution for how often it generates attacks. Those attacks are then represented by attack trees. This is a very elegant way to represent attackers, because it is scalable, flexible, and an accurate representation of real attacker behavior. "Script kiddies" and other less sophisticated attackers have attack trees with very few branches at each level. These attackers have few options, and if they all fail, the attack ends. More sophisticated attackers have trees with much greater breadth, as they have many options available to them. These attack trees are described later in this section.

Figure 5: Adversarial Threat Taxonomy

These attacks will specifically target the confidentiality, integrity, and availability of the user's 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), system policies, and user-level training. Both technical and non-technical attributes are equally important. Initially these attack probabilities will be determined in a subjective manner, through interviews with experts. Developing solid models of these attack-defense relationships remains an open research issue. Several authors have suggested techniques for creating taxonomy of the threat and the modeling of the types of attacks [3, 4, 5, 6, 7].

Several models have been proposed to model the types of attack that an information system may be subjugated with attack trees and petri net modeling methods being the most favorable representation. Attack trees provide a methodical way of describing system security based on the types of attack. [3] An agent-based system may internally represent their possible attack plan based on a forest of possible attack trees where the root of each tree represents a possible goal. A way of modeling these goals may be to represent an attack on the confidentiality, integrity, and availability of a system. For example, Figure 6 may represent the high-level view of an attack agent with the priority of breaking the confidentiality of the system. Each node in the graph represents a set of sub-goals that the agent must achieve 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. [5]

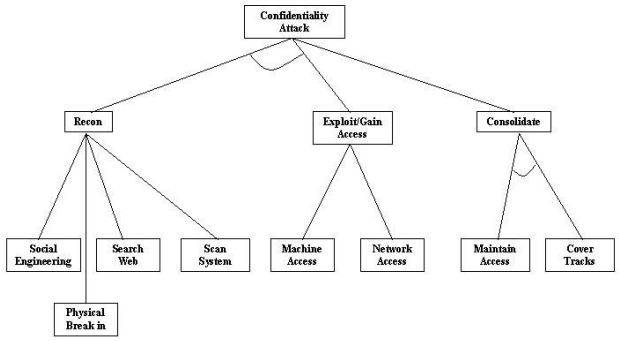

Figure 6: Example Confidentiality Attack

Figure 6 shows an example confidentiality attack. In order for a confidentiality attack to succeed, an attacker has to recon and gain access to the information system. The attack agent may or may not consolidate based on how skilled of an attacker it represents.

As an example, consider an attack on a network that enables an attacker to break the confidentiality of a portion of the system. One way such an attack may occur is through a rouge wireless access point as shown in Figure 7. The attacker first locates wireless access points

through locator software. If the wireless access point (WAP) does not have the wireless equivalent privacy (WEP) encryption, then gaining access to that network is trivial by just configuring the wireless card. Once configured, the attacker can easily sniff the air for any packets transported to the wireless access point from which the agent gained access.

By combining attack trees and the adversarial threat taxonomy, we can model an attack and determine how vulnerable the user's network is. If, in the agent's search through its attack tree, an agent encounters a defensive countermeasure employed by the user, then the agent's attack may fail. Several agents may have different goals or even similar goals, but their attack trees may vary based on their skill/motivation level as determined by the adversarial threat taxonomy. For example, using the confidentiality attack depicted in Figure 7, a "script kiddy" has a higher probability of locating an access point and jumping on an unencrypted access point, but has a lower probability of cracking WEP. Soft factors may be represented with a probability at each node also. An organization that has paid for more training and hired competent system administrators has reduced risk in basic day-to-day tasks thus reducing the probability of rogue access points and access points without WEP or another form of security.

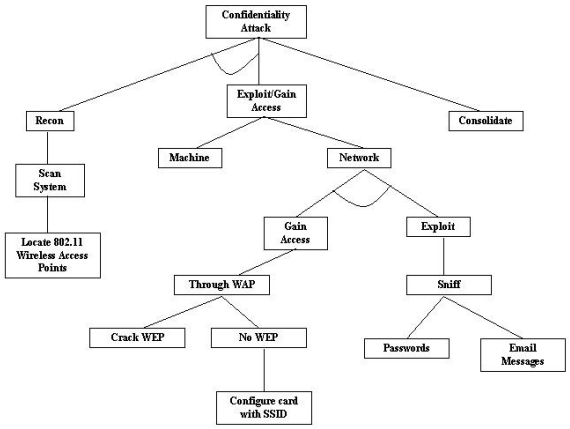

Figure 7: Example Wireless Attack

Defensive modeling involves creating attack trees and then determining where in the system vulnerabilities may exist. In MAADNET a set of finite choices for each possible security tool will be provided to the user. The user must decide what defensive tools to buy (or gather if they are open source) and where in the information system to employ those tools. Tactics such a defense-in-depth, aggressive vulnerability scanning, annual training of users and system administrators, and intrusion response reduce risk, thereby increasing the probability of an attack agent failing in their efforts. The user will have to take into account the type of organization they are defending in order to determine where risk is acceptable. In our initial implementation of MAADNET Competitor, the probability of an attack succeeding will use simple table lookups, taking into account the attributes described previously in this paper; however, we plan on focusing a research effort on better characterizing these relationships in the future.

Scoring a Student's Performance

Since this is both a teaching tool and a decision support tool, metrics must be identified to help rate the goodness of the network (and information assurance) plan. Metrics used in NetBuilder include rating the overall message latency, link percent down time, and node percent down time. These metrics are easily collected and computed. Costs to build, maintain, and repair the network are also easily computed.

MAADNET will also evaluate performance against the confidentiality, integrity, and availability attributes. Recall that the overall scenario file includes a textual description of the scenario. This will give clues to the student about whether confidentiality, integrity, and/or availability are more important to the student's network. The scenario might indicate, for instance, that in the student's organization confidentiality 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 final score given to the student will be computed with the following formula:

$$score = \sum_{i=1}^{n} A_i w_i ,$$

where A_i is the attribute being evaluated (such as throughput, network latency, loss of confidentiality, etc.) and w_i is the importance (or weight) of that attribute in the given scenario. (The sum of all the w_i's is 1.) Recall that the values of the w_i terms are hidden from the student; the student must infer the relative importance of the various attributes from the textual description of the scenario. The final score will be in the range zero to one hundred, where zero is the worst. This score could be used for general feedback to the student, for grading purposes in a networks or information assurance course, or as the basis for determining the winner of a Web-based competition.

Tailoring MAADNET for Pedagogical Purposes

Recall that MAADNET is meant to be a teaching tool. Depending on the course being taught, the students will have different skill sets. In an introductory networks course, the students will know little about network design

and almost nothing about defensive information assurance tools. In an information assurance course, one could expect the students to have a strong networking background (at least at the theoretical level) but perhaps not much information assurance knowledge at the beginning of the semester. Another element of the scenario design is the configuration matrix (also represented in XML), shown in Figure 8. By putting an "X" in a particular cell, the scenario designer is tailoring the educational event to the skill level of the student. Note that the student need not be at the same experience level in all aspects of the simulation.

For those areas in which students are at the Beginner level, MAADNET will offer hints. At the Novice level, fewer hints will be offered; although, the student will have access to some on-line help. At the Expert level, no hints will be offered and there will be little access to on-line help; the user will merely get penalized for poor decisions.

	Hardware Topology, Services, etc.	People	Policies and Procedures	Data	Information Assurance
Beginner					X
Novice		X	X		
Expert	X				

Figure 8: Configure Matrix

Web Delivery

Eventually the MAADNET client and the scenario files will be distributed over the Internet. When a user is happy with his or her network configuration, he or she will submit it (the network configuration is also stored as an XML file) to the server. The MAADNET server will then attach a number of attack agents to the network as specified in a scenario file (hidden from the user) and run the simulation. The efficacy of the student's network configuration will be scored, and the score will be presented to the student. We have explored several technologies for Web delivering MAADNET to students.

The advantage of the applet methodology, regardless of the language used to write the applets, is that anyone with a reasonably recent Web browser has access to the system without downloading any special software. In general the use of applets also permits a more sophisticated interface than the use of scripts. Since the simulation prototype on which MAADNET will be based was written in Java, it seemed natural (but not essential) to use Java for the user interface as well.

In general, Java applets are not allowed to access local resources, such as the file system. Signed applets use a public/private key encryption mechanism to certify that the applet has come from a trusted source. These trusted applets can then be granted limited access to local resources. Applications are not hampered by these requirements. Sun's Java WebStart technology allows a user to download *applications*, vice applets, through a Web browser and run them locally. The other advantage of WebStart is that whenever a user tries to launch the application, WebStart first checks the date of its cached copy of the application against the date of the application at its source. If there is a newer version of the application, it is downloaded and then run; otherwise, the cached version is launched.

CONCLUSION AND FUTURE WORK

When completed, the contributions of the MAADNET project will include an easy-to-use network construction interface, a less-detail oriented traffic model, a service and demand focus to network modeling, the ability to appropriately portray attacks and defenses 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 training and education.

REFERENCES

[1] Saunders, J. H. 2002. The Case for Modeling and Simulation of Information Security. Available online at http://www.johnsaunders.com/papers/securitysimulation.htm

[2] Ressler, S. (2002) West Point Bridge Design Contest. Department of Civil and Mechanical Engineering, USMA. Available online at <http://bridgecontest.usma.edu/>. Last accessed 19 December, 2002.

[3] Uiterwijk, A. (1999). Security Game: Playing for Keeps. Federal Computer Week.

[4] Welch, D. 2002. Adversary Threat Taxonomy. In *Proceedings of the IEEE Information Assurance Workshop*, West Point, NY.

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

[6] Schneier, B. 1999. Attack Trees: Modeling Security Threats. *Dr. Dobb's Journal*.

[7] Moore, A. P., R. J. Ellison, et al. 2001. Attack Modeling for Information Security and Survivability. Technical Note CMU/SEI-2001-TN-001, Software Engineering Institute, Carnegie Mellon University.