MAADNET NetBuilder: A Service/Demand Focused Network Simulator

John M. D. Hill [2], John R. Surdu [1], Scott Lathrop [1], Gregory Conti [2], and Curtis A. Carver, Jr. [2]
[1] Information Technology Operations Center
[2] Department of Electrical Engineering and Computer Science
United States Military Academy, West Point, NY 10996
[John.Hill | John.Surdu | Scott.Lathrop | Curtis.Carver | Gregory.Conti]@usma.edu

Keywords: Network Simulation, Information Assurance, Computer Science Education

Abstract

Network and cyber-defense laboratories are an important part of networking and information assurance education. However, there are rarely enough lab resources or time to allow every student to explore these issues in depth. As par of a larger project, faculty members at West Point are developing MAADNET NetBuilder, a rapid network construction and evaluation tool. Instructors create scenarios defining available facilities, equipment, services, demands that must be supported, and a scoring system. The students construct and configure a network, run the simulation, and receive immediate feedback. A multi-layered design modularizes components, strictly defines interfaces, and simplifies future extensions. At the foundation is a discrete event simulation allowing faster-than-real time execution and a communication layer where traffic is generated for execution and statistics can be captured. The device and link layer defines computing and communication devices and links between them, and is where attributes such as bandwidth are established. The service and demand layer is where mail, file, web, and other services are handled, including the demands placed on those services. The user interface layer allows visual network construction and provides feedback from execution of the simulation. MAADNET NetBuilder is a simulation-based approach to network instruction that should prove useful to anyone engaged in network and security education.

BACKGROUND

Networking and information assurance are important parts of the curriculum in the Computer Science (CS) program at the United States Military Academy (USMA) at West Point, a Center of Excellence for Information Assurance (IA) Education. Designing, building, and evaluating networks require time and equipment. There are several network and computer laboratories available for instruction, including an isolated Information Warfare (IWAR) lab used for the CyberDefend exercise [1]. Designing IA exercises, constructing the network, and preparing for a distributed attack/defense competition, is even more complex.

Although the students gain a lot from their experiences in the networking course and the information assurance course, there is little time and precious few resources available for individual exploration of network construction and management issues. There is a clear need for a mechanism that allows students to rapidly prototype a network configuration, experiment with alternate configurations, and receive immediate feedback. Once comfortable with network construction, students need to be able to configure security mechanisms and observe their effectiveness against attacks.

In response to this need, faculty members in the CS program and researchers in the Information Technology and Operations Center (ITOC) have started the Military Academy Attack/Defense Network (MAADNET) project [2]. The ultimate goal of this project is to provide a distributed virtual attack/defense playground in which teams can compete against each other. The foundation of the project is the MAADNET NetBuilder, a visually oriented simulation-based network construction prototyping tool.

RELATED WORK

Several networking design and simulation packages already exist. Examples include OPNET IT Guru and OPNET Modeler, NetCracker, NetRule, and Predictor. [3, 4, 5, 6, 7] These tools focus on the design of network architectures that move individual data packets at different layers within the OSI protocol stack. Such simulations allow users to predict the impact on application performance and scalability based on underlying changes to network protocols. With NetBuilder as the underlying framework, MAADNET focuses on services and demands, aggregating communication flows at the message level.

Others have built simulation based-tools for Information Warfare (IW) or Information Assurance (IA) education. Tools such as InfoChess and CyberProtect portray a mix between higher-level IW issues to low-level network security concerns. InfoChess focuses on high-level topics such as psychological operations, military deception, and electronic warfare. [8, 9] CyberProtect takes into account some “soft” evaluation metrics such as purchasing of computer hardware and software, computer security tools, and training. [10] The user designs their network by dragging and dropping purchased nodes onto their network and then likewise placing security tools. The system is then subjugated to an "attack" by scripted scenarios with an evaluation result provided to the user. The MAADNET designers are currently exploring the possibility of collaborating and leveraging the work completed by the developers of CyberProtect. The intended audiences for these IW/IA simulations are military information operation planners, managers of information systems, or system administrators with the focus of their instruction being primarily on the management of resources rather than the technical and tactical details of how to employ a secure information system.

NETBUILDER OVERV IEW

The networking design and simulation tools and the IW/IA tools don't strike the balance between the granularity required in building a secure network and the higher level, soft issues associated with assuring information processed, stored, and transported on the system desired by the MAADNET team. Using MAADNET NetBuilder as a basis, the MAADNET project is an attempt to find the right balance in order to serve as an educational tool for students, faculty, managers, and administrators of information systems.

MAADNET uses a layered design (see Figure 1). This simplifies construction of the system, and makes it easier to expand it later. Because NetBuilder is the foundation of the entire MAADNET project, extensibility for defense and attack mechanisms is a prime development consideration.

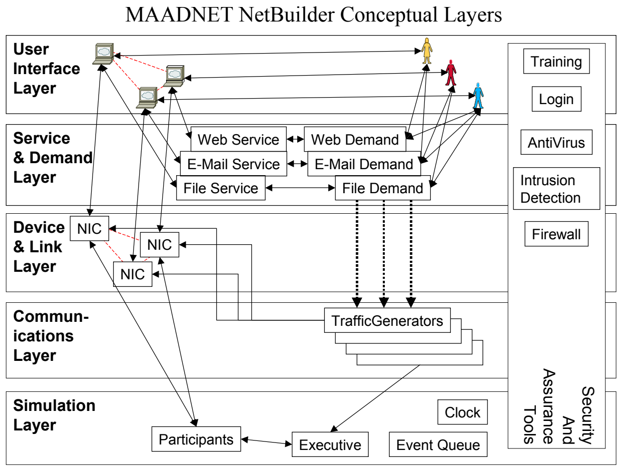

Figure 1: Conceptual Layers in MAADNET NetBuilder

MAADNET NetBuilder is designed to let the user rapidly construct an IT system using aggregations representing standard underlying TCP/IP protocols and standard system security tools (such as firewalls, intrusion detection systems, and vulnerability scanners), without having to worry about the intricacies of what happens at the physical, data link, and networking link layers of the OSI reference model. Basically, if the devices can connect, the user is allowed to connect them. If traffic can flow, an aggregation model represents the flow.

SCENARIOS, CONFIGURATIONS, AND SCORING

In the MAADNET system, scenarios describe an environment (physical space), the available devices, links, services, and support personnel, and the demands (associated with users) that will be placed on the system. They also define conditions that must be satisfied (mail services available to all, etc.). An important part of the scenario (particularly in the future competition aspect of MAADNET) is the definition of costs and other trackable attributes, and the scoring criteria based on those attributes. MAADNET scenarios are specified in separate eXtensible Markup Language (XML) files. For example, the environment file specifies the physical layout of the problem, like the shape and size of the rooms.

Instructors define scenarios to teach or reinforce particular concepts (saturation, for example). The user then designs and deploys the network, emplaces services, and determines policy and procedures. Within the constraints of an initial budget, the user can choose (buy) the hardware to meet the requirements, choose (buy) security tools, and select (hire) the people to administer the system. The configured system can then be subjected to a series of simulated operations, including attacks from a spectrum of attackers to include enthusiastic script kiddies to cyber terrorists. As MAADNET becomes more sophisticated, levels of user network and IA awareness will be represented.

SIMULATION LAYER

The foundation of the layer system is a classic discrete event simulation (DES) architecture. The simulation executive registers entities (simulation participants) and processes requests from those entities to schedule events (such as message arrival times at a device). The events are maintained in temporal order in an event queue. A clock is run in various multiples of real time. At the appropriate time, the executive removes each event from the queue then notifies the scheduling entity that the event has occurred and must be executed. In the process of executing events, other events may be scheduled. For instance, a traffic generator might generate a message and then schedule the next message generation event.

With the configurable parameters for representing demands (see the Service and Demand Layer section), one can richly define different types of users in an organization, including admin staff, engineers, Web developers, programmers, etc. When users are added to the system, their messages generators are added as simulation participants. When a message generation event is pulled from the event queue, it points to a particular traffic generator. The message generation event calls a method in the traffic generator that creates a new message. Part of generating a message is the determination of the path through the system from the sender to the receiver(s). Once the messages are created the traffic generator schedules two more events:

- It determines when the message will reach its first hop along the path to the destination and schedules a message arrival event at that time.
- It determines when its next message will be generated and schedules a new message generation event.

In NetBuilder, other types of events are possible, such as equipment failure events and equipment repair events. The various workstations, routers, switches, servers, hubs, and other hardware items all have probabilities distributions that define their mean time between failures (MTBF) and mean time to repair (MTTR). The quality and quantity of system administrators may effect MTBF and MTTR. When the simulation is initialized, each piece of hardware determines stochastically when they will fail and schedules a hardware failure event. When that hardware failure event is executed, the MTTR is used to determine when the device will be repaired, and a hardware repair event is scheduled. Between the time that the hardware failure event is executed and the time that the hardware repair event is executed, the device is considered inoperative.

When NetBuilder is extended to include the information assurance simulation capabilities of MAADNET, this same event-driven methodology will be used to schedule attacks and counteractions.

COMMUNICATIONS LAYER

The communication layer is the interface between the underlying simulation and the devices and links. Conceptually, it is in this layer that the traffic is generated and passed up to the device and link layer. It is important to reiterate that traffic in the NetBuilder system does not represent packets, and is not managed with packet-level protocols. Rather, demands in the Service and Demand layer cause traffic generators in the Communications Layer to send messages. Said messages come with bandwidth requirements, and may have other attributes, such as priorities.. This aggregation is sufficient to determine bandwidth utilization in devices and across links, and to capture statistics for feedback and analysis.

DEVICE AND LINK LAYER

A device is any hardware that participates in the network. This includes computers, routers, network interface cards, etc. A link is any means of conveying traffic between devices. This includes unshielded twisted pair wire, coaxial cables, and even wireless links. Devices and links have defined capabilities, including max bandwidth, number and types of ports, etc. Devices and links are modeled in an object hierarchy that, among other advantages, allows for easy addition of new device or link types.

When the user runs the system, the demands cause message traffic to start flowing. The messages are scheduled into the simulation, with arrival times at devices based on their bandwidth requirement versus the bandwidth available. The interface provides visual feedback of bandwidth bottlenecks, and a score based on criteria defined in the scenario. With this feedback, the user reconfigures the network in an attempt to improve performance.

SERVICE AND DEMAND LAYER

NetBuilder has models for several kinds of common services, such as file, mail, and web services. As other services are required, the model can be extended. Services placed on compatible devices respond to demand traffic by producing their own traffic, and placing a load on their devices. The services file specifies the list of services that must be provided to the users. A snippet of the services file is shown in Figure 2.

```
<services>
  <service>
    <type>email</type>
    <class>EmailServer.class</class>
    <name>Outlook_Eggspress</name>
   </service>
  <service>
    <type>web</type>
    <class>Webserver.class</class>
    <name>Arapaho</name>
   </service>
  <service>
    <type>db</type>
    <class>DatabaseServer.class</class>
    <name>Hot_Fission</name>
   </service>
  <service>
    <type>file</type>
    <class>DatabaseServer.class</class>
    <name>Phyle_Service</name>
   </service>
</services>
```

Figure 2: Portion of the Services File

Note that in the case of services, this file identifies the Java class associated with the implementation of that service within the simulation. This makes it very easy to create a new Webserver subclass that behaves differently than the base class and then use it in MAADNET.

In NetBuilder, demands are associated with people (icons) in the system, and include the loads they place on mail, file, web, and other services. Part of the defined scenario is a collection of such people. Associated with each person is a number of traffic generators that generate different types of messages at defined intervals. These intervals can be constant or generated from one of the common families of probability distributions. Users have traffic generators for Web access, database access, access to file servers, CPU usage, and Email. Email is further divided into three categories: Email to a single user within the organization, Email to all users within the organization, and Email to the Internet. Figure 3 is an example of a user who only sends Email.

```
<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

The NetBuilder user must provide sufficient services for said demands, and ensure there is enough computing power and bandwidth to support them. This will involve making decisions about purchasing hardware and services, and in how to configure the network. Placement of people in the system (since the demands they create have different impacts depending on which part of the system they are associated with) is now a significant issue as well.

SECURITY AND ASSURANCE TOOLS

The tools file includes XML descriptions of those protocols, devices, personnel, and software applications such as firewalls, intrusion detection systems, etc. that are available to the users of the simulation. An example tools file that describes network devices is shown in Figure 4: The nature of each tool dictates which layer of the conceptual model it operates in. For example, a hardware firewall would operate in the device and link layer while an intrusion detection system would operate in the Services and Demands layer (recall Figure 1).

```
<network_devices>

  <component>
    <class>Router.class</class>
    <type>router</type>
    <name>Sysko_5000c</name>
    <picture>Sysko_5000c.jpg</picture>
    <numCards>6</numCards>
    <cpu>2.9</cpu>
    <bufferSize>5000</bufferSize>
    <cost>1900</cost>
  </component>

  <component>
    <class>Workstation.class</class>
    <type>workstation</type>
    <name>Binford_9000_Workstation</name>

<picture>Binford_9000_Workstation.jpg</picture>
    <cpu>1.2</cpu>
    <memory>256</memory>
    <disk>40</disk>
    <cost>2000</cost>
  </component>

  <component>
    <class>EthernetCard.class</class>
    <type>networkcard</type>
    <media>100baseT</media>
    <name>Schlockworks_7_Ethernet_Card</name>
    <picture>EthernetCard.jpg</picture>
    <cost>40</cost>
  </component>

  <component>
    <class>WireLink.class</class>
    <type>networkcable</type>
    <media>100baseT</media>
    <name>network_cable</name>
    <picture>network_cable.jpg</picture>
    <cost>2</cost>
  </component>
```

Figure 4: Portion of the Network Devices File

USER INTERFACE LAYER

The user interface provides visual network construction, with devices, services, demands, etc., dragged off of selection bars generated from the scenario. A foundation holds rooms that serve as the drop point for device placement, and links are established with drag gestures from one device to another. If linking rules are not satisfied, the drag is not allowed to succeed. Services are set up by simply dropping them on a compatible device. Demands are established by placing people (with their associated demand profile) on a compatible device (usually a workstation). A control panel starts, pauses, rewinds, and stops the simulation. The display panel provides visual feedback on bandwidth utilization, bottleneck identification, costs, and other attributes. The visual drag and drop interface and control enables rapid configuration and re-configuration and provides immediate feedback, allowing the user to rapidly get the answer to “what-if” questions.

FUTURE WORK

Students will first use the NetBuilder prototype as a tool in a networking course. Beyond the educational objectives, the primary goal will be to figure out what works right, what needs improvement, and what enhancements need to be added. Once the modifications are made, the more robust version will serve as the foundation for the construction of the attack/defense pieces and the competition.

Attack modeling considers the threat and the possible types of attacks. In MAADNET attacks will consist agents for each type of threat., with a forest of attack trees. The attacks will target the confidentiality, integrity, and availability of the user's information system, with the probability of an attack succeeding 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 user-level training. Development of solid models of attack/defense relationships will incorporate several suggested techniques for creating a taxonomy of the threat and the modeling of the types of attacks. [11, 12, 13, 14, 15]

Welch breaks attackers into a taxonomy based upon two factors: their skill (*enthusiast, minstrel, virtuoso, composer, maestro*) and motivation (*explorer, delinquent, activist, investigator, criminal, agent, cyberwarrior*). [11] For example, a typical script kiddie 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. They are the subjects of much of the work done in the HoneyNet Project. [16] At the opposite extreme in the taxonomy is the *maestro cyberwarrior*, perhaps an individual, state-sponsored, or sponsored non-governmental (NGO) terrorist organization.

Attack trees and petri net modeling methods are popular attack representations. Attack trees provide a methodical way of describing system security based on the types of attack. [13] The attack plan is based on a forest of possible attack trees where the root of each tree represents a possible goal. 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 (all must be achieved) or an OR-decomposition (at least one must be achieved). [14] an example confidentiality attack tree appears in Figure 5.

By combining attack trees and the adversarial threat taxonomy, we can model an attack and determine how vulnerable the user's network is. In the agent's search through its attack tree an agent may encounter a defensive countermeasure employed by the user, causing the agent's attack to fail. Each agent 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. 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 compromise.

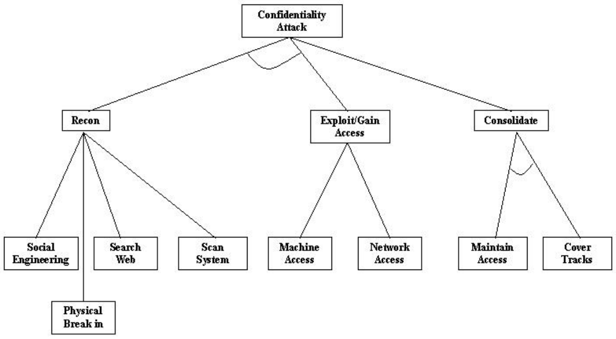

Figure 5: Example Confidentiality Attack Tree

Defensive modeling involves creating attack trees and then determining where vulnerabilities in the system may exist. In MAADNET, a set of finite choices for each possible security tool will be provided to the user. The user will decide what tools to buy (or gather, if they are open source) and where in the information system to employ those tools. Employment of tactics such as a defense-in-depth, aggressive vulnerability scanning, annual training of users and system administrators, etc., reduce risk and increase the probability of an attack agent failing in their efforts. The user will have to the entire system configuration, services, policies, etc., in order to determine acceptable risks.

CONCLUSIONS

MAADNET NetBuilder is designed to be a very valuable tool for instruction in networking and information assurance. The most valuable conclusions will come out of student experiences with the NetBuilder prototype. The improvements based on the assessment of those experiences will serve to improve not only NetBuilder, but also the entire MAADNET system that will be built upon it.

REFERENCES

[1] Jackson, W. (2002). Cadets Keep NSA Crackers at Bay. Government Computer News. **21**.

[2] Curtis A. Carver, J., J. R. Surdu, et al. 2002. Military Academy Attack/Defense Network. In *Proceedings of the 3rd Annual IEEE Information Assurance Workshop*, West Point, NY.

[3] OPNET IT Guru. OPNET Technologies. Available online [accessed September 15, 2002] at <http://www.mil3.com/products/itguru/home.html>.

[4] OPNET Modeler. OPNET Technologies. Available online [accessed September 15, 2002] at <http://www.mil3.com/products/modeler/home.html>.

[5] NetCracker Architecture. NetCracker Technology Corporation. Available online [accessed September 15, 2002] at <http://www.netcracker.com/architecture.html>.

[6] NetRule: The First Practical tool to Predict network Performance and Plan Successful Changes. Analytical Engines, Inc. Available online [accessed September 15, 2002] at <http://www.netrule.com>.

[7] Predictor: WAN Provisioning and Growth Management. Compuware Corporation. Available online [accessed September 15, 2002] at <http://www.compuware.com/products/vantage/predictor/>.

[8] Saunders, J. H. 2002. Simulation Approaches in Information Security Education. In *Proceedings of the Sixth National Colloquium for Information System Security Education*, Redmond, Washington.

[9] InfoChess Online: An Interactive Competitive Strategy Game. Aegis Research Corporation. Available online [accessed September 15, 2002] at <http://www.aegisresearch.com/products/p_infochess.html>
.

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

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

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

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

[14] 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.

[15] McDermott, J. P. 2000. Attack Net Penetration Testing. In *Proceedings of the 2000 Workshop on New Security Paradigms*, Cork, Ireland, 15-21.

[16] Project, H. 2002. *Know Your Enemy: Revealing the Security Tools, tactics, and Motives of the Blackhat Community*. Boston: Addison Wesley.