*Proceedings of the 2007 Winter Simulation Conference*
*S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds.*

# CYBER ATTACK MODELING AND SIMULATION FOR NETWORK SECURITY ANALYSIS

Michael E. Kuhl
Jason Kistner
Kevin Costantini

Industrial & Systems Engineering Department
Rochester Institute of Technology
Rochester, NY 14623, U.S.A.

Moises Sudit

National Center for Multisource Information Fusion
University at Buffalo
Buffalo, NY 14260, U.S.A.

## ABSTRACT

Cyber security methods are continually being developed. To test these methods many organizations utilize both virtual and physical networks which can be costly and time consuming. As an alternative, in this paper, we present a simulation modeling approach to represent computer networks and intrusion detection systems (IDS) to efficiently simulate cyber attack scenarios. The outcome of the simulation model is a set of IDS alerts that can be used to test and evaluate cyber security systems. In particular, the simulation methodology is designed to test information fusion systems for cyber security that are under development.

## 1 INTRODUCTION

As the use of computer networks grows, cyber security is becoming increasingly important. To enable systems administrators to better protect their networks, cyber security tools are employed to warn of suspicious network activity. In some situations, systems administrators have to deal with millions of such warnings each day. Consequently, situational awareness and threat assessment tools that employ information fusion techniques are being developed to aid in fighting cyber attacks. As these systems are being developed, data is needed to test and evaluate their performance. As an alternative to a physical computer network, a simulation modeling methodology is presented. The simulation method allows the user to construct a virtual computer network that produces cyber attack warnings representative of those produced by intrusion detection systems. Consequently, this flexible simulation modeling framework will enable the efficient generation of data to test and evaluate situational awareness and treat assessment tools for cyber security.

There is some research in modeling of computer networks and cyber attacks. For example, Lee et al. (2004) and Nicol et al. (2003) present simulation modeling methods for simulating computer network traffic at the packet level. Although simulating the flow and processing of packets in the computer network is possible (potentially billions of packets per day), only a small fraction of the packets cause alerts to be produced by the intrusion detection system which in turn would be used by the information fusion tools. Furthermore, modeling a system at this level of detail requires great amounts of time and effort for modeling as well as requiring large amounts of computer processing time for simulating "good" packets. As an alternative to modeling the details of packet flow in a network, this work presents a simulation model for simulating the behavior of the intrusion detection system by producing simulated alerts representative of malicious cyber attacks and non-malicious network activity based on the user's specification. Consequently, the user can efficiently construct scenarios of various computer networks and cyber attacks and generate the corresponding alerts.

## 2 BACKGROUND AND RELATED WORK

This work is based in the need for testing situational awareness tools that are being developed to detect and analyze attacks on computer networks. Since conducting cyber attack experiments on computer systems that contain critical data is very undesirable, several alternatives have been used. One alternative consists of setting up a physical computer network absent of any critical data, performing cyber attacks on the network, and collecting data from intrusion detection systems. A second alternative consists of generating synthetic data through the use of simulation.

These two approaches have varying degrees of requirements, capabilities, and limitations. The physical computer network requires the physical machines, networking, and IDS components. Consequently, conducting experiments on various network configurations involving different machines, servers, routing systems, IDS sensors, etc. requires reconfiguration of the network and setting up the network to produce the desired network activity and cyber attacks. The advantage of using the physical network

is that the data produced is from a real network as opposed to an abstract representation. This also has some disadvantages in that it is impossible to replicate the experiment exactly (if so desired) and the data produced is difficult to validate to ensure all desired information is accounted for in the ground truth. Since physical networks are not perfectly reliable, data can be missed, processed incorrectly, etc.

The simulation approach requires knowledge of the operation of the desired network and its operation. This information must be captured by the simulation model to represent the behavior of the network. However, as discussed briefly in the introduction, the level of detail included in the model will depend on the goal of the simulation. In this case, the packet level information and computer network traffic details are not needed, so the simulation can be constructed at a higher level to produce alerts caused by cyber attacks and harmless network traffic. Once the framework of the model has been established, various network configurations can be efficiently created and experiments can be conducted with various attack scenarios. Since the simulation experiments are controlled, they can be repeated exactly and all ground truth information is known.

## 3 OVERVIEW OF THE SIMULATION MODEL

A discrete-event simulation model has been developed for generating representative cyber attack and intrusion detection sensor alert data. Although the model is primarily designed to be used in testing cyber situational awareness and analysis tools, other applications such as training of systems analysts may also make effective use of the model. The simulation model is initially implemented in the ARENA simulation software. An object-oriented model written in Java is currently under development. Although this paper utilizes the ARENA model to illustrate the modeling concepts, the focus is on the concepts themselves.

The simulation model provides a user with the ability to construct a representative computer network and setup and execute a series of cyber attacks on certain target machines within that network. IDS sensors that are setup within this network produce appropriate alerts based on the traffic they observe within the network. The alerts produced consist of a combination of the alerts produced as a result of attack actions and as a result of typical "noise" (non-malicious network traffic that triggers an alert.)

Figure 1 displays an example network interface setup using the ARENA model. To effectively model a network setup in ARENA and to provide users that may not have extensive simulation training with a friendly interface, custom modules were created for the network devices. The simulated computer networks consist of three primary types of devices: machines, connectors, and subnets.

A machine can represent an individual computer or server. Machine characteristics can be specified including the IP address, the operating system, and the type of IDS sensor on the machine (if any). For each IDS sensor specified, an associated output file will be generated containing the sequence of alerts produced when the simulation is run.

A connector represents the means by which computers are connected, such as through a switch or a router. The network connectivity plays in important role in establishing the path that an attacker can take through the network. The connector also has network IDS sensors that can be represented which are used to monitor any network traffic that travels through the connector and produce alerts corresponding to known potentially harmful actions.

A subnet represents a group of several machines with connectivity to the network that all share a common set of properties (such as the operating system). Machines within a subnet contain the same set of properties that could be specified if the machines were placed into the network individually. The subnet just provides an efficient method of specifying groups of computers (particularly useful when specifying large networks.)

Connector lines are used in the model to connect the modules and represent the connection of machines/subnets to a connector, as well as the connections between connectors themselves.

When a computer network has been created, an attack scenario can be setup and run on the network. An attack scenario consists of a series of specified cyber attacks occurring over a period of time along with a specified quantity of network noise. A user-interface with a series of forms is used to specify the desired scenario. The model structure enables manual or automatic attack generation. In the manual mode, the user can specify all of the details of the attack scenario including the sequence and timing of attack actions as well as the path the attack will take through the computer network. In the automatic mode, the user can specify the goal (ultimate attack action and target computer) of the attack, and the simulation model will generate a random, feasible sequence of attack actions along a path that leads to the goal. Additional parameters that represent the behavior of the attacker can also be specified. These parameters include the efficiency, stealth, and skill of the attack being modeled. The efficiency refers to how direct the attack is, and this utilizes a range between 0 and 1, with 1 representing the most efficient attack path. The stealth parameter refers to how well the attack avoids detection, primarily by avoiding intermediate "goal" steps, and this also utilizes a range between 0 and 1. The skill refers to the probability of success for each step.

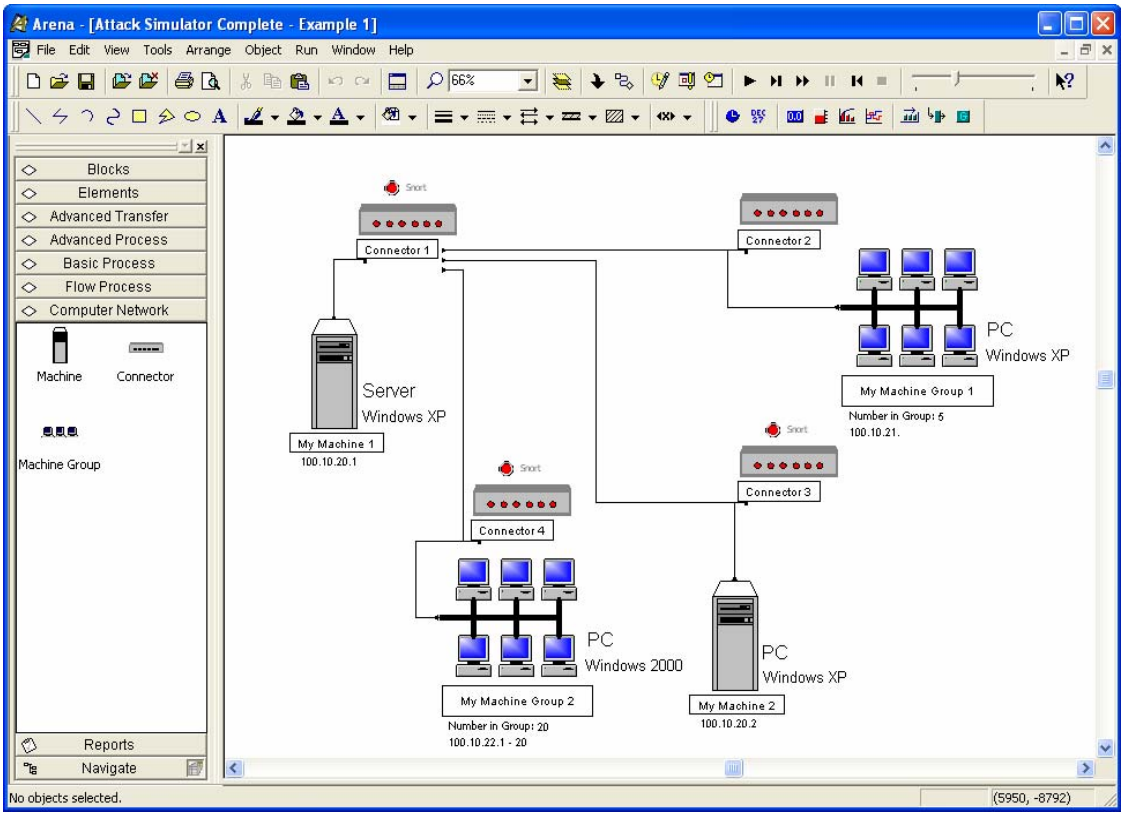


Figure 1: Sample network interface in arena model

Currently, an attack scenario in the ARENA model can handle up to 25 attacks with 250 steps per attack. Also, for each type of attack, the user can specify the time between attack steps based on a fixed number or on a random number sampled from an exponential distribution (with a specified mean). The steps/actions available for use in an attack are chosen from a categorized list of 2,237 known exploits in 5 major groups and 23 subgroups. If no specific exploit is selected, one will be chosen at random based on the subgroup. In addition to attacks, the user can specify, the rate at which non-malicious traffic alerts (noise) is generated, as well as the probability of noise alerts corresponding to each of the action categories.

Once the scenario has been created, the information is saved in a file for future use. The simulation is then run, and the attack scenario is executed. The output of the simulation includes a file listing the actions generated for each attack (known as the "ground truth") and the time the action occurred. In addition, an output file containing IDS alerts is produced for each IDS sensor specified in the modeled network. These files containing IDS alerts are intended to be used to test the situational awareness and analysis tools.

## 4 SIMULATION METHODOLOGY

This section discusses in detail the general approaches taken in modeling computer networks, modeling cyber attacks, and simulating cyber attacks and generating corresponding IDS data.

### 4.1 Modeling Computer Networks

As described in the previous section, the computer network is modeled using two basic constructs: machines and connectors. The third construct, subnets, represents a group of machines. The modules representing the machines, connectors, and subnets provide a visual representation of the computer network. However, functionally, these modules provide a logical method for the user to enter the data about the computer network including whether the machine can be accessed externally from the Internet. The connecting lines showing the connectivity of the network are used to construct a from-to type of matrix representing the network topology that will be used in the attack generation. The details of the devices (such as the type of IDS) are stored as variables that can be accessed based on the device ID. The devices used can be easily modified by

double-clicking their corresponding representation in the interface to bring up a form to enter or change information.

### 4.2 Modeling Cyber Attacks

The scope of this work is on cyber attacks that are initiated by a hacker through the Internet. Although insider attacks could also be modeled, this is not the primary purpose of the model. The progress that a hacker can make in an attack is dependant upon the hacker's capabilities and the vulnerabilities of the network. The methods for modeling and simulating the initiation and progression of cyber attacks through a computer network included in this model are based on Sudit et al. (2005).

Sudit et al. (2005) place the sequence of attack actions that a hacker may use into stages that correspond to the hacker's capabilities given the current state of the network. These stages are referred to as Stage 0 through Stage 9 where Stage 0 represents generally reconnaissance activities on the external part of the computer network where the attacker is using exploits to simply gain more information about the network. (In this discussion an external machine is one that can be accessed from the Internet, and an internal machine is a machine that can only be accessed from an external machine through a firewall or from another internal machine.) Stage 0 – Stage 4 represent hacker actions on external machines, and Stage 5-Stage 9 represent hacker actions on internal machines. Table 1 list some typical hacker actions that correspond to an attack stage.

The hacker can attack an organization's machine that is on the external side of the computer network. Once the external machine has been successfully compromised, the hacker can use the compromised external machine to work their way through the external network until the capability to access internal machines is reached. Once the hacker has infiltrated the internal network, the internal machines can be compromised until the hacker reaches their goal. Figure 2 illustrates the cyber attack process from the internet to a goal on an internal machine.

Table 1. Typical hacker actions in a cyber attack

| Stage | Typical Action |
|---|---|
| 0 | Recon. Footprinting |
| 1 | Intrusion User |
| 2 | Escalation Service |
| 3 | Intrusion Root |
| 4 | Goal Denial of Service |
| 5 | Recon. Enumeration |
| 6 | Intrusion User |
| 7 | Escalation Service |
| 8 | Intrusion Root |
| 9 | Goal Pilfering |

The simulation model includes automated and user-specified cyber attack generation methods. The automated method utilizes the network specifications and connectivity in combination with a guidance template of the available stages to determine the capabilities of the attacker and vulnerabilities of the network and generate a feasible sequence of attack steps for the cyber attacks. The graph-based guidance template is used to determine which groups of actions are feasible at different points of the attack. A diagram of the graph-based template that the simulation model currently operates under is shown in Figure 3 (S0, S1, …, S9 represent Stage 0, Stage 1, … , Stage 9.)

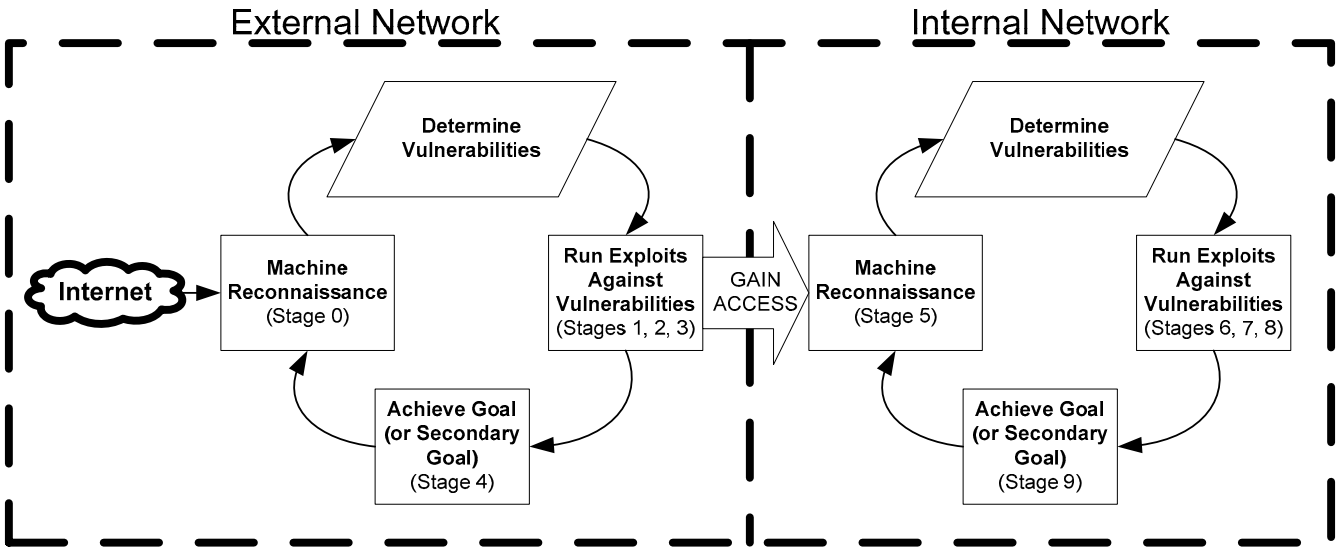


Figure 2. Progression of a cyber attack on a computer network from the internet

The graph is a directed graph, which means that an edge (arc) only indicates a feasible transition in the direction that the edge is pointing. Nodes within the same group form a complete graph in which each node is connected to the every other node. This graph-based template is represented as an adjacency matrix of 1's and 0's representing which stages are accessible after which other stages have been performed.

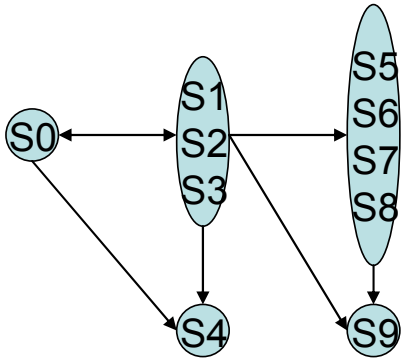


Figure 3. Directed graph representing attack structure

Given the attack structure (in the form of the guidance template) and the network configuration specified, the user also specifies a target machine, a goal, and several other attack related parameters (discussed previously) through a series of forms. Figure 4 illustrates the automated method that is used to generate the specific multi-stage attack.

In generating the steps (prior to simulating them over a time period), the methodology works backwards through the network by first defining the attack's target and finding a path up out of the network that the hacker could attack through. The logic first chooses an attacker (machine from which the hacker could execute the attack step) which is able to communicate with the chosen target based on the topology of the network. After the attack progression for the current target is determined, a new target can be chosen. The options for the new target are a machine with which the current attacker can communicate or the current attacker itself. Choosing the current attacker as the new target will move the attack to a higher level of the network topology (toward the external machines) to model the way in which hackers penetrate a network. If the chosen target is not the current attacker, the logic will repeat the steps for determining guidance template progression and determine another target, using Stage 5 through Stage 9. However, if the current attacker is chosen for the new target, the attack generation moves up a level in the network topology. Thus, the logic evaluates whether the chosen target has become an external machine. If the chosen target is not an external machine, the logic will choose a new attacker who can reach the new target and repeat the attack generation process. However, if the target is external, the attacker must be attacking from the Internet. Thus, the attacker IP address for attacks on external machines is created randomly since hackers will generally "spoof," or disguise, their IP address when attacking from the Internet. The logic will then determine the guidance template progression for the external target, now using Stage 0 through Stage 4.

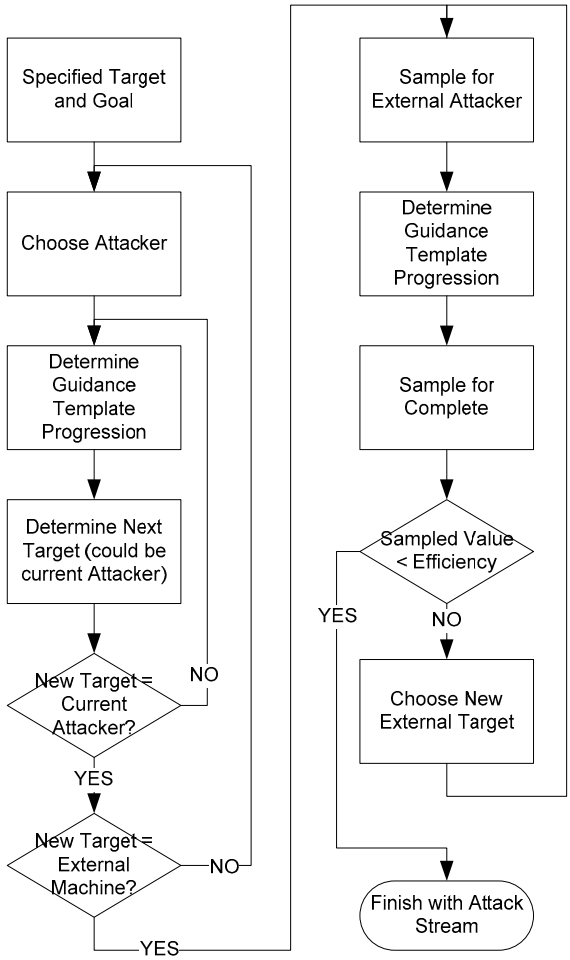


Figure 4. Automate attack generation method

When the guidance template progression is determined, the logic will sample a random value between zero and one and evaluate this value against the efficiency factor specified by the user at the beginning of the simulation run. If the sampled value is greater than the efficiency factor, a new target will be chosen and the attack generation steps repeated. If the sampled value is less than the efficiency factor, the attack generation is complete.

If the user prefers to specify the specific steps of the attack, various levels of automation are provided down to attacks that can be fully specified by the user.

### 4.3 Simulating Cyber Attacks and Generating IDS Sensor Alert Data

The general modeling approach for representing the cyber attacks is to model the individual attacks (or the hackers executing the attacks) as entities. One entity is created at the beginning of the simulation to represent each attack. Each entity is assigned a unique attack identification number. Then the entity executes the appropriate code that samples the necessary attack information. Each entity representing an attack stores information about the first step of the attack in its defined list of attributes. The attack information about the first attack step. Then the entity is delayed until the first step in the attack is specified to start. This delay can be constant or can be sampled randomly from an exponential distribution depending on the user's specification. Finally, the entity is routed to the station corresponding to the target IP address for the first attack step.

A generic station sub-model represents each of the machine locations in the computer network. When entities (attacks) are routed to the station, the attack step is executed. The success of the attack step is evaluated by sampling from a uniform distribution on (0,1) and comparing this number with the skill parameter (or probability of success) defined for the attack. If the step fails, the necessary attack step information is sampled, the attack information is assigned to attributes, the target IP station is determined, the entity is delayed, and the entity is then routed to the appropriate station similar to the sequence of actions executed above. If the attack step succeeds, the attack step number is incremented by one, and the next step in the attack is executed. Depending on the result, the appropriate alert information is written to output files. This process is repeated until the last step in the attack is executed successfully. At this point, the number of completed attacks is tallied, and the entity is disposed.

The simulation model generates both attacks and noise. The noise represents IDS alerts produced by ordinary network activity. The occurrence rate of noise alerts are specified by the user, and generated via a Poisson arrival process.

The simulation dynamically produces several output files. These files include ground truth files for the attack action and IDS alerts, and IDS alert files. The Ground Truth Actions file contains a listing of the hacker attack actions that were executed during each attack as well as an indication of whether each action was successful or not. The ground truth files for IDS alerts are produced for each type of IDS that is used in the system and contains all of the alert information corresponding to the actual attack actions and excludes any noise alerts. Finally, the IDS alert files that are produced include formatted alerts which are dependent on the location of the IDS in the network. The IDS alert files include alerts produced from both attack actions and noise and are representative of the information that a system administrator may receive when using IDSs to monitor network activity.

## 5 CYBER ATTACK EXAMPLE

In this section, an example attack scenario on a computer network is presented. The scenario makes use of the automatic attack generator to create two separate attacks

### 5.1 Attack Descriptions

The network diagram is shown in Figure 8. The goal of the first attack is to create a backdoor on a machine in the BPN Group subnet, while the goal of the second attack is to perform pilfering on an machine in the Research Group 1 subnet. For both attacks, information is gathered about the external network, and then the VPN server is penetrated. The server is then used as a stepping stone to reach the target machine. Since the automatic attack generator is used, the attack steps are not entered by the user, but instead are generated during the simulation.

The computer network created using the Cyber Attack Simulator is shown in Figure 6. A summary of the network is as follows:

1 main web-server;
3 main subnet domains;
2 subnet domains have further subnets attached;
Only one external machine; and
Red dots indicate IDS sensor presence.

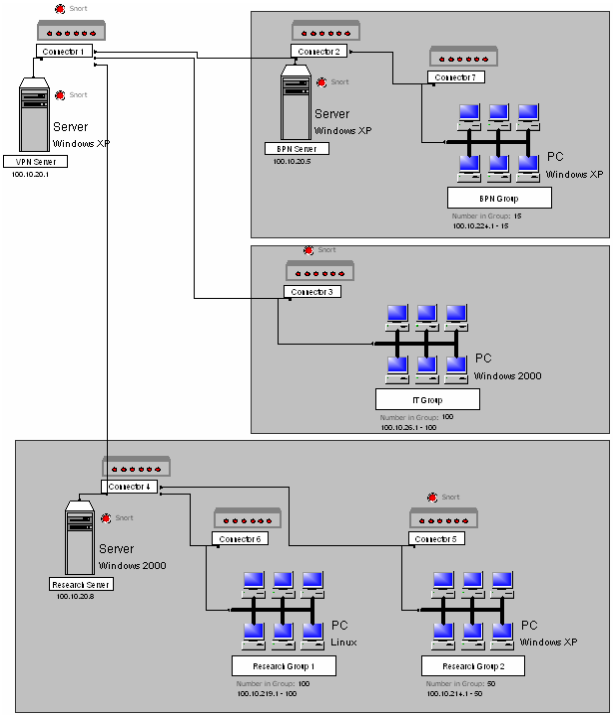


Figure 6. Sample Network

Table 2 illustrates the attack information provided via the auto-attack user interface. The scenario has 150 noise alerts per hour on average where 85% of the noise is reconnaissance, 10% is escalation, and 5% is classified as miscellaneous. Also, the simulation will run for five additional minutes after the last attack is complete. This simulation run results in the generation of the steps shown in Table 3, considered the ground truth. The steps are sorted by the time at which they occur.

The combination of these attack steps and the noise within the network create a large number of IDS sensor alerts during the simulation run. Three sample alerts produced are displayed in Figure 7.

Table 2. Auto attack parameters

| Attack | Target | Goal Type | Efficiency | Stealth | Skill | Delay | Step Time |
|---|---|---|---|---|---|---|---|
| Attack 1 | 100.10.224.11 | Backdoor | 0.9 | 1.0 | 1.0 | 2 | 3 |
| Attack 2 | 100.10.219.41 | Pilfering | 0.6 | 0.8 | 0.9 | 5 | 4 |

Table 3. Auto attack steps generated

| Attack | Step | Group | Subgroup | Action/Exploit | Source IP | Target IP | Success? |
|---|---|---|---|---|---|---|---|
| 1 | 1 | Recon | Enumeration | WEB-FRONTPAGE rad fp30reg.dll access | 211.21.49.174 | 100.10.20.1 | SUCCESS |
| 2 | 1 | Recon | Footprinting | SCAN SOCKS Proxy attempt | 237.136.23.194 | 100.10.20.1 | FAIL |
| 2 | 1 | Recon | Footprinting | RPC portmap admind request UDP | 173.231.24.107 | 100.10.20.1 | SUCCESS |
| 2 | 2 | Intrusion | User | WEB-CGI tcsh access | 104.28.71.164 | 100.10.20.1 | SUCCESS |
| 1 | 2 | Escalation | Service | EXPLOIT x86 Linux samba overflow | 16.203.97.119 | 100.10.20.1 | SUCCESS |
| 1 | 3 | Intrusion | Root | WEB-CGI htmlscript attempt | 100.10.20.1 | 100.10.26.87 | SUCCESS |
| 1 | 4 | Intrusion | User | WEB-COLDFUSION application.cfm access | 100.10.20.1 | 100.10.20.5 | SUCCESS |
| 1 | 5 | Misc | Other | DNS EXPLOIT x86 Linux overflow attempt (ADMv2) | 100.10.20.5 | 100.10.224.11 | SUCCESS |
| 1 | 6 | Goal | Backdoor | BACKDOOR Doly 2.0 access | 100.10.20.5 | 100.10.224.11 | SUCCESS |
| 2 | 3 | Escalation | OS | NETBIOS SMB DCERPC ISystemActivator bind attempt | 100.10.20.1 | 100.10.26.45 | SUCCESS |
| 2 | 4 | Escalation | OS | NETBIOS DCERPC Remote Activation bind attempt | 100.10.20.1 | 100.10.20.8 | SUCCESS |
| 2 | 5 | Escalation | OS | NETBIOS DCERPC Remote Activation bind attempt | 100.10.20.1 | 100.10.20.8 | SUCCESS |
| 2 | 6 | Intrusion | Root | WEB-CGI psunami.cgi access | 100.10.20.8 | 100.10.219.98 | SUCCESS |
| 2 | 7 | Escalation | OS | NETBIOS SMB DCERPC Remote Activation bind attempt | 100.10.219.98 | 100.10.219.43 | SUCCESS |
| 2 | 8 | Intrusion | Root | WEB-PHP TextPortal admin.php default password (12345) attempt | 100.10.219.98 | 100.10.219.63 | SUCCESS |
| 2 | 9 | Misc | Other | POLICY FTP MKD possible warez site | 100.10.219.98 | 100.10.219.63 | FAIL |
| 2 | 9 | Misc | Other | POLICY FTP MKD possible warez site | 100.10.219.98 | 100.10.219.63 | SUCCESS |
| 2 | 10 | Intrusion | User | WEB-MISC Domino bookmark.nsf access | 100.10.219.98 | 100.10.219.21 | SUCCESS |
| 2 | 11 | Escalation | Service | FTP XCWD overflow attempt | 100.10.219.98 | 100.10.219.21 | SUCCESS |
| 2 | 12 | Intrusion | Other | WEB-PHP shoutbox.php directory traversal attempt | 100.10.219.98 | 100.10.219.83 | SUCCESS |
| 2 | 13 | Misc | Other | POLICY FTP MKD possible warez site | 100.10.219.98 | 100.10.219.83 | SUCCESS |
| 2 | 14 | Misc | Other | POLICY FTP CWD possible warez site | 100.10.219.83 | 100.10.219.67 | SUCCESS |
| 2 | 15 | Escalation | Service | EXPLOIT ebola USER overflow attempt | 100.10.219.83 | 100.10.219.67 | SUCCESS |
| 2 | 16 | Goal | Dos | SMTP Content-Transfer-Encoding overflow attempt | 100.10.219.83 | 100.10.219.41 | SUCCESS |
| 2 | 17 | Goal | Pilfering | ORACLE truncate table attempt | 100.10.219.83 | 100.10.219.41 | SUCCESS |
| 2 | 18 | Goal | Pilfering | ORACLE truncate table attempt | 100.10.219.83 | 100.10.219.41 | SUCCESS |

```
06/02-15:15:01.958499  [**] [1:905:4] WEB-COLDFUSION application.cfm access [**] [Classification: attempted-recon] [Priority: 2] \{TCP\} 100.10.20.1:781 -> 100.10.20.5:594/par
06/02-15:16:49.364106  [**] [1:265:4] DNS EXPLOIT x86 Linux overflow attempt (ADMv2) [**] [Classification: attempted-admin] [Priority: 1] \{TCP\} 100.10.20.5:632 -> 100.10.224.11:248/par
06/02-15:17:54.826833  [**] [1:119:4] BACKDOOR Doly 2.0 access [**] [Classification: misc-activity] [Priority: 3] \{TCP\} 100.10.20.5:756 -> 100.10.224.11:798/par
```

Figure 7. Sample snort alerts produced

## 6 UTILIZING THE ATTACK SIMULATOR TO EVALUATE INFORMATION FUSION METHODS

Information fusion is the process of associating, correlating, and combining data and information from single or multiple sources to estimate parameters, characteristics, and behaviors of a system for the purposes of analysis or decision support (Linus, 2001). Figure 8 illustrates the application of information fusion to a system. The ground truth is the actual status of the system. From the ground truth, a set of data or information can be sensed and passed to an information fusion process. The fused information is passed to a decision maker that may take some action on the system in attempt to change the system status.

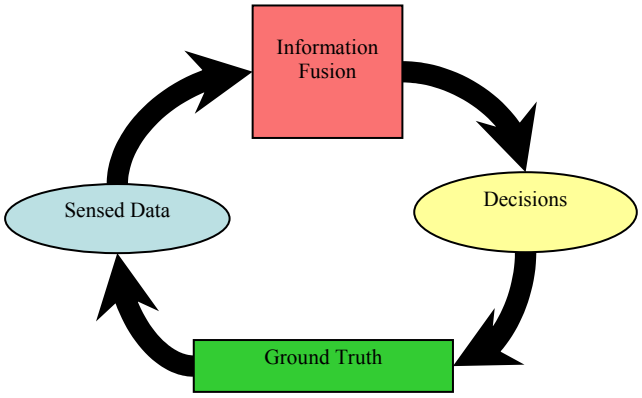


Figure 8. Information fusion applied to a system

Some of the most difficult aspects of developing information fusion methods are validation and evaluation. The validation and evaluation processes both require data for testing and experimentation. In some cases, the systems to which the information fusion process is to be used are readily available so direct experimentation can take place. However, in many applications the systems for which the information fusion processes are being designed do not exist, may be destructive, or may be cost prohibitive to set up. In these cases, simulation provides a good alternative.

For example, in the context of cyber security, situational awareness and threat assessment tools including information fusion techniques are being developed to aid systems administrators in identifying and analyzing cyber attacks on computer networks (Sudit et al. 2005). These tools work by primarily processing alerts produced by intrusion detection systems (sensors) on the computer network. To test and evaluate these tools, physical computer networks have been set up to perform experiments from which data is collected. As an alternative, a simulation modeling method and software is developed to generate synthetic data.

The simulation modeling methodology is a first step at presenting a flexible modeling framework that will easily allow a user to specify the configuration of the computer network under study and efficiently generate cyber attacks. The output that the simulation model gives, represents the actual alerts that a system administrator would see in their daily duties. Consequently, there is great potential for continued development of this model to cyber applications ranging from network evaluation to training.

## 7 CURRENT DEVELOPMENT

Current work entails the development of an object-oriented Java simulation model. The primary motivation behind this development is to create a simulator that is platform independent and easier to use for individuals with expertise in computer networks and cyber security rather than simulation. This new model improves upon the ARENA model by providing several features allowing for networks and attacks to be defined in more detail and allowing for a wider range of inputs to and outputs from the model. These features include:

- Allowing multiple attack scenarios to be created and saved with a network;
- Separating the auto-attack generation and event simulation, and providing a display for each;
- Defining a list of services running on a machine;
- Defining a list of ports/protocols that are allowed or banned through a specific connector path;
- Utilizing the machine vulnerabilities and connector attributes to determine the selection and the success of an action/exploit (as opposed to strict probability);
- Allowing network traffic to be routed through more than two connectors (based on connector link attributes); and
- Exporting a modeled network to a “Virtual Terrain” XML file or importing a network into the model from a “Virtual Terrain” XML file (The concept of a “Virtual Terrain” is used to represent networks in the Information Fusion realm).

Ongoing work entails the continued validation and the addition of features to enhance the model to represent cyber attacks on computer networks as accurately as possible.

## 8 CONCLUSIONS

The Cyber Attack Simulator presented in this paper is capable of generating IDS alert and ground truth files based on the specification of a computer network and attacks. The simulator is built with a user interface to allow the creation of various computer network configurations and

attack actions. The model also incorporates a method for automated attack generation given the network configuration, characteristics describing hacker capabilities, and vulnerabilities of the network.

## REFERENCES


Lee, J.-S., J.-R. Jung, J.-S. Park, and S. D. Chi. 2004. Linux-based system modeling for cyber attack simulation. In *Proceedings of the 13th International Conference on AI, Simulation, and Planning in High Autonomy Systems*, Jeju Island.

Linus, J. 2001. An Introduction to Data and Information Fusion. (Presentation) Available Online via <http://www.infofusion.buffalo.edu/tutorialPage.php> [Accessed July 15, 2007]

Kelton, W. D., R. P. Sadowski, and D. T. Sturrock. 2004. *Simulation with ARENA*, Third Edition, McGraw-Hill, Boston, MA.

Nicol, D., J. Liu, M. Liljenstam, and G. Yan. 2003. Simulation of large-scale networks using SSF. In *Proceedings of the 2003 Winter Simulation Conference*, ed. S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, 650-657. Institute of Electrical and Electronics Engineers, Piscataway, NJ.

Sudit, M., A. Stotz, and M. Holender. 2005. Situational awareness of coordinated cyber attack. In *Proceedings of the International Society for Optical Engineering Conference*, Orlando, FL.


## ACKNOWLEDGMENTS

This work was sponsored in part by the U.S. Air Force Research Laboratory in Rome, NY. Special thanks for their valuable insights and contributions goes to Jay Yang from RIT and our collaborators at the National Center for Multisource Information Fusion and CUBRC. Special thanks also goes to Katie McConky and Greg Tauer for their efforts in the development the Java-based attack simulator.

## AUTHOR BIOGRAPHIES

**MICHAEL E. KUHL** is an Associate Professor in the Industrial and Systems Engineering Department at Rochester Institute of Technology. He has a B.S. in Industrial Engineering from Bradley University (1992), M.S. in Industrial Engineering from North Carolina State University (1994) and a Ph.D. in Industrial Engineering from North Carolina State University (1997). His research interests include simulation modeling methodologies with application to cyber security, healthcare, and semiconductor manufacturing, and simulation analysis procedures for input modeling and output analysis. He served at the Proceedings Editor for the 2005 Winter Simulation Conference. He is vice-president of the INFORMS Simulation Society, and a member of IIE and ASEE. His e-mail address is <Michael.Kuhl@rit.edu> and his web address is <www.rit.edu/~mekeie>.

**JASON KISTNER** is a Master of Science student in the Industrial and Systems Engineering Department at Rochester Institute of Technology. He is a member of IIE. His e-mail address is <jpk7781@rit.edu>.

**KEVIN COSTANTINI** is a Master of Science student in the Industrial and Systems Engineering Department at Rochester Institute of Technology. He is a member of IIE. His e-mail address is <kcc3121@rit.edu> .

**MOISES SUDIT** is managing director of the National Center for Multisource Information Fusion at the University at Buffalo. In this capacity he has lead a number of research programs related to Information Fusion and related topics. Dr. Moises Sudit's primary research interests are in the theory and applications of Discrete Optimization. More specifically, he has been concerned in the design and analysis of methods to solve problems in the areas of Integer Programming and Combinatorial Optimization. One primary goal of this research has been the development of efficient exact and approximate (heuristic) procedures to solve large-scale engineering and management problems. Dr. Sudit received his Doctorate Degree in Operations Research from Purdue University, his Masters of Science in Operations Research from Stanford University and his Bachelor of Science in Industrial Engineering from the Georgia Institute of Technology.