Interactive Simulation Tools for Information Assurance Education

H. Yu, K. Williams, J. Xu, X. Yuan, *B. Chu, *B. Kang, *T. Kombol
Department of Computer Science
College of Engineering
North Carolina A&T State University
* Department of Software Engineering and Information System
University of North Carolina Charlotte

Abstract

In this paper we present several interactive education tools that demonstrate information assurance concepts, firewall configuration and functions, LAN attack and prevention. These tools have been used in the Department of Computer Science at North Carolina A&T State University for security related courses to let students get hands-on experience and increase student interest and confidence in Information Assurance (IA) knowledge and skills. In order to improve the instructional capability of IA faculty nationwide these tools were distributed at the 2008 Faculty Development Workshop on Cyber Games and Interactive Simulation. A workshop survey was conducted and received excellent results. The participants of the workshop will integrate these tools to implement their own IA curriculum during academic 2008-2009.

1. Introduction

There is growing evidence that incorporating hands-on exercise components increases students' interest in Information Assurance (IA) studies and enhances their learning experience [1, 3, 7, 9, 12, 13). Interactive simulation is highly interactive hands-on exercises and helps students understand abstract IA concepts through interactive explorations and visualizations. However, hands-on experiences in traditional computer science curricula have been limited to programming assignments. Developing new types of hands-on exercises and using them in IA education become necessary.

In order to enhance information assurance education, faculty members in the Department of Computer Science at North Carolina A&T State University (NC A&T SU) have developed several interactive education tools to demonstrate Information Assurance concepts, firewall configuration and function, LAN attack and prevention. These tools have been used to teach information assurance related courses in NC A&T SU. In order to increase the capacity of current IA faculty members nationwide, these tools were introduced at the 2008 Faculty Development Workshop on Cyber Games and Interactive Simulations. All materials related with these tools were available for the workshop attendees.

In the second part of this paper four interactive simulation tools that demonstrate IA concepts will be introduced. Firewall configuration and function will be presented in section 3. A LAN attack on a switched network simulator will be discussed in section 4. The results of the workshop and assessment will be presented in section 5. The conclusions will be given in section 6.

2. Interactive Simulation Tools for IA Concepts

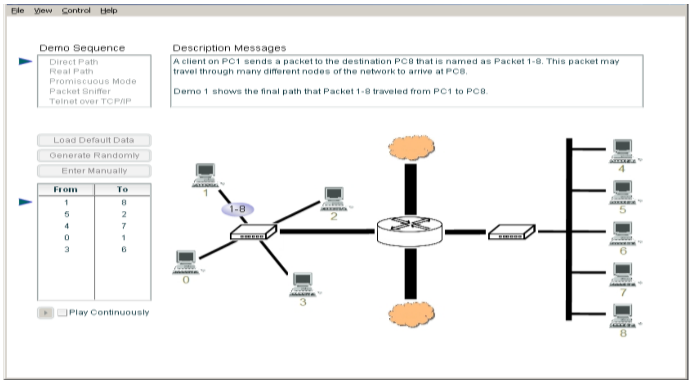
Four interactive simulation tools, which are a packet sniffer simulator, an animated learning tool for Kerberos authentication architecture, a visualization tool for wireless network attacks, and an interactive SYN flood simulator, are developed. These tools let students get hands-on experience and help them learn IA concepts.

2.1. Packet Sniffer Simulator

The packet sniffer simulator [11] was designed to demonstrate how a packet sniffer works progressively. It consists of a suite of five demos: direct path, real path, promiscuous mode, packet sniffer and Telnet Over TCP/IP. The first four demos progressively demonstrate how a packet sniffer works at a higher level.

The fifth demo depicts how a data packet is transmitted in the network at a more in-depth level. It demonstrates a protocol stack and animates the encapsulation and de-encapsulation process.

The user interface that Demo I to IV use (Figure 1) includes four components. The “Demo Sequence” lists the names of the five demos that the user can select to play; The “Description Message” briefly describes the animation; The “Network Architecture” includes two subnets with star and bus topologies respectively connected by a router. The “Simulation Data” allows the user to interact with the simulator through changing the input data.



Description Messages

A client on PC1 sends a packet to the destination PC8 that is named as Packet 1-8. This packet may travel through many different nodes of the network to arrive at PC8.

Demo 1 shows the final path that Packet 1-8 traveled from PC1 to PC8.

From	To
1	8
5	2
4	7
0	1
3	6

Figure 1 A snapshot of the packet sniffer Demo I

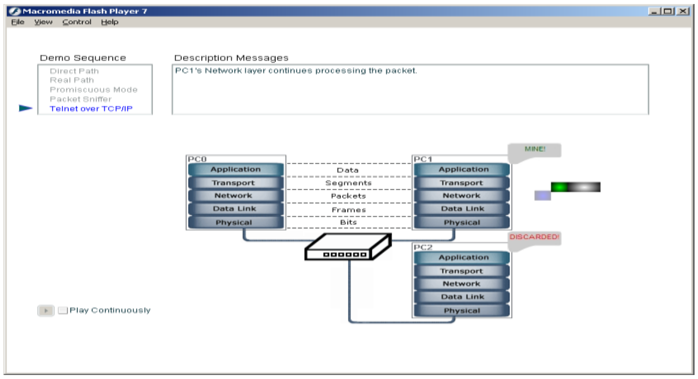
Demo I “The Direct Path” displays the path a data packet goes through to reach the destination. A data packet is represented as an oval, labeled by the source and destination numbers. Figure 1 shows a data packet moving from computer 1 to the hub to the left. It will go through the router in the middle, the hub to the right, and finally arrive at computer 8. Since hub is used in the two subnets, the real path that Packet 1-8 traverses is not the same as the direct path.

Demo II “The Real Path” demonstrates the real path of a data packet in the simulated network. In Demo II, a data packet is broadcasted to all the computers in the subnet, and is forwarded by the router to the other subnet. All computers in the subnet will receive the packet but only the destination computer will accept the data packet.

Demo III “The Promiscuous Mode” demonstrates the fact that, when a computer’s network interface card is configured into promiscuous mode, the computer will accept all incoming frames without checking the destinations of the frames.

Demo IV “The Packet Sniffer” demonstrates that when a computer’s network interface card is configured into promiscuous mode, and it has packet sniffer installed, it will accept all incoming frames, and also examine the content of the frames according to the configuration of the packet sniffer.

Demo V “Telnet Over TCP/IP” displays a protocol stack and animates the encapsulation and de-encapsulation process assuming a Telnet application sending data packets over a network with TCP/IP protocol. Figure 2 represents three computers (PC0, PC1 and PC2) connected to a hub. A protocol stack of five layers are displayed in each computer. The animation demonstrates a data packet generated at the application layer at PC0 being encapsulated while moving down through the protocol stack, and being de-encapsulated while moving up through the protocol stack at PC1 and PC2. PC2 discards the data packet since its destination address it not PC2, while PC1 accepts the packet and the frame eventually reaches the application layer.



PC1's Network layer continues processing the packet.

Figure 2. A snapshot of the packet snifer Demo V: The De-Encapsulation Process

2.2. An Animated Learning Tool for Kerberos Authentication Architecture

Kerberos has become one of the most widely used services. However, it is very difficult for the students to understand the many elements contained in the protocol. The animated learning tool for Kerberos authentication architecture is developed to assist students in understanding this protocol [12]. The design of this tool is based on the strategy used by Bill Bryant of Project Athena. It includes a series of four animated scenes that progressively demonstrate the ideas that underlie the design of the Kerberos authentication architecture. Each scene demonstrates a stage of the protocol development. Each successive scene adds additional complexity to counter security vulnerabilities revealed in the preceding scene. For some of the scenes, hacking scenarios are provided to illustrate the vulnerabilities of the scene. This software tool is based on Kerberos version 4. “Challenge” questions are provided for each scene to test the student’s understanding of the scene, and guide the student in grasping the main points demonstrated by the animation.

Scene I, “Distributed Authentication”, shows an open network with three workstations (Alice, Bob and John) and two service servers (Microsoft Exchange/Email Server and Windows NT application/file server). Alice and Bob are connected to a hub; the hub, John and the two servers are connected to a switch. Each service server has a password database to authenticate the users requesting service. Scene I animates the following process: Alice sends a request which includes Alice’s ID and password to the email server, the email server verifies Alice using its password database and sends a response back to Alice if Alice is verified. The drawbacks of distributed authentication are: (1) The user’s ID and password are sent in plaintext over the network; (2) If a user wants to change his password, he has to change passwords in all the servers.

Scene II, “Centralized Authentication”, demonstrates the authentication process with an Authentication Server (AS). Alice sends to the AS her ID, password and the email server ID; the AS then verifies Alice using the centralized password database. If Alice is verified, it creates an email server ticket encrypted with the email server’s key and sends it to Alice. Alice then sends the email server ticket along with her ID to the email server. The email server decrypts the ticket, verifies Alice and sends the requested information back to Alice. The Centralized Authentication has the following drawbacks: (1) The client’s ID and password are still sent in plaintext over the network; (2) The service server ticket is reusable and can be stolen; (3) The client has to give the AS his password every time he wants to use a service for which he does not have a ticket. Two hacking scenarios are designed for this scene: password stealing with a packet sniffer, and replay attack.

Scene III, “Ticket Granting Service”, adds a Ticket-Granting Server (TGS) based on Centralized Authentication. The client first sends a message to the authentication server to request a ticket granting ticket. After acquiring the ticket granting ticket, the client sends a request to the TGS for a service ticket. After verifying the client and the ticket granting ticket, the TGS sends to the client the service ticket. The client then uses the service ticket to request service from the service server. With this mechanism, the client only has to use his password once. The password is not sent over the network in clear text. However, the ticket can still be stolen, and replay attack can be launched. Figure 3 shows a snapshot of scene III.

Scene IV, “Kerberos System”, shows the actual Kerberos protocol. It counters the vulnerability of replay attack by using session keys and authenticator. The Kerberos protocol also allows mutual authentication, that is, the servers may be required to authenticate themselves to the clients.

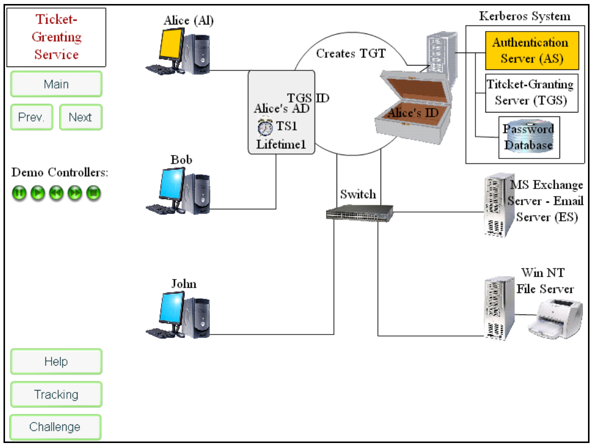


Figure 3. A snapshot of scene III: Ticket Granting Service

2.3. A Visualization Tool for Wireless Network Attacks

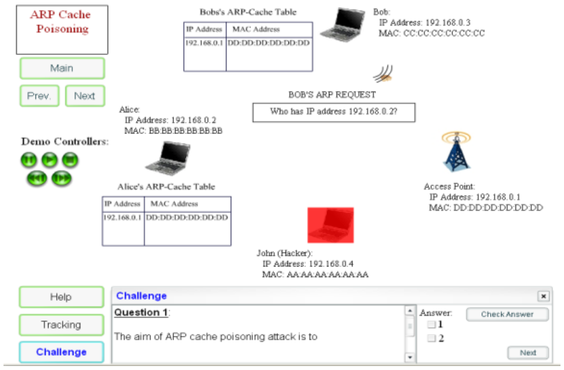
The visualization tool for wireless network attacks [13] animates the following attacks popular in wireless networks: Eavesdropping, Evil Twin, Man in the Middle, ARP Cache Poisoning, and ARP Request Replay. The tool includes a series of five demos that visualize five attack scenarios. Each demo includes at least one user (Alice), sometimes a second user (Bob), a hacker (John), and an access point. The tool also provides “challenge questions” to give the user a quiz on the animation he/she watched. In what follows, the visualization scenarios of the five types of wireless network attacks are described.

The Eavesdropping scenario demonstrates how a hacker eavesdrops the communication between two wireless nodes. The network includes two users (Alice, Bob), an AP, and a hacker (John). John sets his network interface card to promiscuous mode. Alice sends a message to Bob, the packets are sent from Alice to the AP, then forwarded by the AP to Bob. John captures the message between Alice and the AP.

The Evil Twin scenario demonstrates how the hacker sets up an evil twin access point, and has the client connect to it. The hacker John eavesdrops the communication between Alice and the AP. Alice sends a PROBE REQUEST for the ESSID of a nearby AP. The AP responds with a PROBE RESPONSE which includes its ESSID and BSSID. John captures the ESSID information in the PROBE RESPONSE. John sets up a Rogue AP using the ESSID John captured through eavesdropping. John then broadcasts a de-authenticate frame to Alice to disconnect Alice from the AP. Alice re-associates with the rogue AP since it is physically closer.

The Man in the Middle scenario demonstrates how an attacker sets up a rogue AP and intercepts message between a user and an AP. Alice is connected to the AP. John eavesdrops the communications between Alice and the AP, and sets up a Rogue AP. John broadcasts a de-authenticate frame to Alice to disconnect Alice from the AP. Alice re-authenticates and re-associates to John; John authenticates and associates to the AP on behalf of Alice.

The ARP Cache Poisoning scenario demonstrates how the hacker causes incorrect IP/MAC address mapping to be added to a computer’s ARP cache. Figure 4 shows a snapshot of the ARP Cache Poisoning demo. In Figure 4, each user computer is labeled with its IP and MAC addresses. An ARP Cache table is displayed beside each user computer. The ARP cache table stores the mappings of IP addresses to MAC addresses.



Bobs's ARP-Cache Table

IP Address	MAC Address
192.168.0.1	DD:DD:DD:DD:DD:DD

Bob:
IP Address: 192.168.0.3
MAC: CC:CC:CC:CC:CC:CC

BOB'S ARP REQUEST

Who has IP address 192.168.0.2?

Alice:
IP Address: 192.168.0.2
MAC: BB:BB:BB:BB:BB:BB

Alice's ARP-Cache Table

IP Address	MAC Address
192.168.0.1	DD:DD:DD:DD:DD:DD

Access Point:
IP Address: 192.168.0.1
MAC: DD:DD:DD:DD:DD:DD

John (Hacker):
IP Address: 192.168.0.4
MAC: AA:AA:AA:AA:AA:AA

Challenge

Question 1:

The aim of ARP cache poisoning attack is to

Answer:
□ 1
□ 2

Figure 4. A snapshot of the ARP Cache Poisoning demo

In the scenario, Bob sends an ARP request: "Who has IP address 192.168.0.2"?, which is broadcasted by the AP to Alice and John. Alice sends out ARP response: "I have 192.168.0.2, my MAC address is: BB:BB:BB:BB:BB:BB", and Alice's IP and MAC addresses are added to Bob's ARP cache table. John then sends to Alice a fake ARP response: I have 192.168.0.3, my MAC address is AA:AA:AA:AA:AA:AA. Alice's ARP cache table adds the entry 192.168.0.3, AA:AA:AA:AA:AA:AA. Alice sends a packet to Bob; because of the incorrect IP and MAC address mapping, the package is sent to John instead of Bob. John then forwards the packet to Bob, acting as a man in the middle between Alice and Bob.

The ARP Request Replay scenario demonstrates how an attacker replays ARP request in order to crack WEP key. In the scenario, Each station is labeled with its IP Address and MAC Address. John also has an IV table that stores the IVs captured by the hacker.

First Bob broadcasts an ARP request: "Who has IP address 192.168.0.2"? John captures Bob's ARP request. Alice sends the encrypted ARP response to Bob. The encrypted ARP response includes an IV. John captures the ARP response through eavesdropping, extracts the IV from the packet, and stores the IV in the captured IV table. John then resends the captured ARP request: "Who has IP address 192.168.0.2"? and Alice sends the encrypted ARP response with a new IV to John. John captures the new IV. The above process is repeated until John collects enough different IVs for cracking the WEP key.

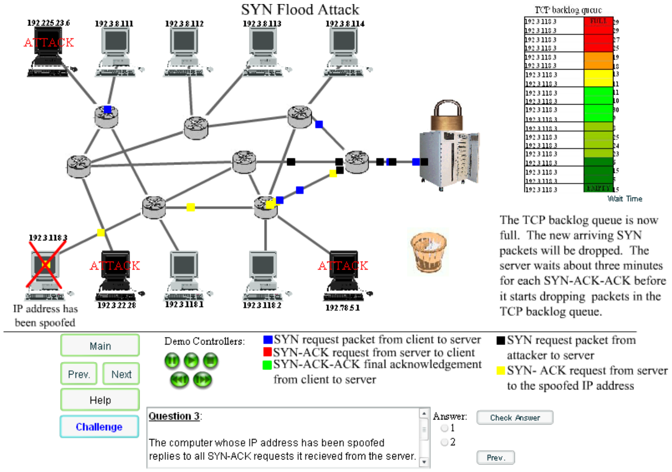
2.4. Interactive SYN Flood Simulator

SYN flood attack is a type of Denial of Service [4, 6]. The process of requesting a webpage consists of a TCP three-way handshake. The SYN flood simulator is developed to demonstrate the concepts of normal network traffics, how SYN flood occurs, and using firewall to prevent some SYN flood attacks. It allows students interact with the simulator and take challenge questions. The first demonstration is normal network traffic, which displays how the TCP three way handshake works. The second one is a SYN flood attack, which displays how the two way handshake occurs as well as what happens during a SYN flood attack. Prevention method is the last demonstration, which displays a firewall as a prevention method to a SYN flood attack.

The SYN Flood animated demo page provides four different options they are *Normal Network Traffic, SYN Flood Attack, Prevention Method and HOME.* Once a user clicks on any button, except the home, a brief description of that demonstration will appear in a window in the middle of the screen along with a corresponding *Start Demo* button, which will run the demonstration.

The *SYN flood attack* demonstration exhibits of how an actual SYN flood attack occurs and what happens during that time period (see Figure 5). This demonstration simulates ten computers, one server, and one TCP backlog queue. Three computers, whose color is black, represent attackers. One computer, marked with a red x, represents its IP address has been spoofed. The server is the targeted computer, the TCP backlog queue stores all received SYN request with their IP addresses. The wait time is the lifetime of each packet since it was received by the server, and waits for a final acknowledgement from the client.

During the SYN flood attack, a combination of attackers and normal computers begin to make requests to establish a connection to the server. Attackers will begin sending out a large number of half open SYN packets, using a spoofed source IP address, to make a request to connect to the server. The packet color of the attacker's SYN request packet is black**.** Once the server receives the request it will send out a SYN-ACK request to the spoofed IP address and wait for its response, which it will never receive. The packet color changes to yellow. Each request will be stored in the TCP backlog queue and will expire when its wait time runs out. For this demo the wait time is located next to each packets request on the TCP backlog queue. At the same time the regular computers will begin making requests to connect to the server as well. Packet color is blue. The TCP backlog queue will become full since it's trying to process request faster than it can handle them. At this time, for the demonstration, a trash can and a lock will appear. The lock represents the TCP backlog queue is full therefore no new SYN request can be accepted. The trash can represent some of the packets being dropped. It shows access being denied because the TCP backlog queue is full. Once the wait time of each packet, which is thirty-two seconds for this demonstration, runs down the SYN packet will be removed from the TCP backlog queue. The new arriving packets will be accepted.



The TCP backlog queue is now full. The new arriving SYN packets will be dropped. The server waits about three minutes for each SYN-ACK-ACK before it starts dropping packets in the TCP backlog queue.

🟦 SYN request packet from client to server
🟥 SYN-ACK request from server to client
🟩 SYN-ACK-ACK final acknowledgement from client to server
■ SYN request packet from attacker to server
🟨 SYN- ACK request from server to the spoofed IP address

Question 3:

The computer whose IP address has been spoofed replies to all SYN-ACK requests it recieved from the server.

Answer:
○ 1
○ 2

Figure 5. SYN flood Attack Demonstration

In the prevention method demonstration, a firewall is placed in between the last router and the server as a prevention method for the SYN flood attack. Firewalls are very useful against SYN flood attacks but they cannot completely stop SYN flood attacks. Firewalls help slow down SYN flood attack, filter out packets that are sent in a half open state and increase server efficiency. In this demonstration the firewall only drops some of the half open packets, which are sent from attackers, but not all. The packets that are filtered out by the firewall are dropped at random. This demonstration is similar to the SYN flood attack. Attackers will begin sending out a large number of half open SYN packets, using a spoofed source IP address, to make a request to connect to the server. Once the server receives the request it will send out a SYN-ACK request to the spoofed IP address and wait for its response, which it will never receive. Each request will be stored in the TCP backlog queue and wait for the client's response. These packets will stay in the TCP backlog queue until its waiting time expires. In this demonstration some attack request packets will pass the firewall and some won't. There is a trash can which represents attack packets that have been filtered out and did not pass through the firewall.

3. Firewall Simulator

Firewalls are a security tool that can be implemented in many environments to provide perimeter protection. Most commercial networks install a dedicated firewall device between their local intranet and the Internet to monitor traffic entering and leaving the network. To be effective, firewalls have to be properly configured. Proper configuration is not immediately obvious or static. It can be difficult to differentiate a good data packet from a potentially damaging packet.

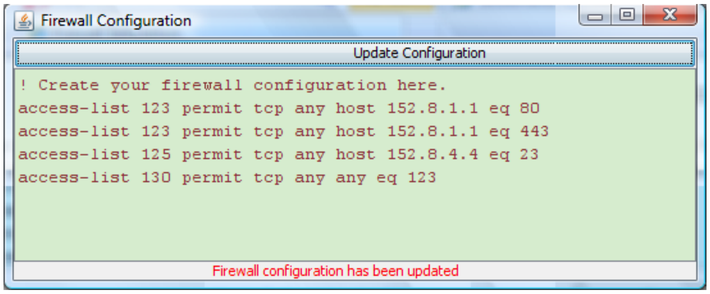
Teaching students how to properly configure a firewall can be challenging. Most schools do not have commercial firewall systems available for the students to use nor do they have the capability to generate reasonable benign and attack traffic. To assist in teaching students how to configure a firewall, we created a firewall simulator. This interactive learning tool allows students to configure a virtual firewall to protect a virtual network. The goals of our firewall simulator are:

- Teach the student how to configure a firewall according to a given set of requirements. The requirements change during the simulation to mimic the real world.
- Provide a realistic interface for configuring the firewall. In addition to learning the concepts of firewalls, the students will be come familiar with how to use commercial firewalls. Our simulator uses a syntax similar to that used by Cisco firewalls [8].
- Be interactive and engaging to provide the students with a simulated hands-on learning environment.
- Prevent the trivial solution of configuring the firewall to stop all network traffic from being a successful strategy.
- Be fun. If the students enjoy using the simulator, they will be much more likely to use the tool and learn the material. Our firewall provides a competitive environment where each student protects their own virtual network while taking action against the networks of other students.

Students using the firewall simulator assume the role of a network administrator who is responsible for configuring a firewall between the local virtual network and the Internet. After a student starts the simulator, they can begin modifying the configuration of their firewall to meet the initial requires: allow traffic to their imaginary corporate web server, allow email to the email server and allow external traffic to access their DNS server. Figure 6 shows the firewall configuration window with Cisco-like syntax. Later in the simulation the instructor monitoring the simulation can add further configuration requirements such as allowing instant message traffic (two different formats), add an ftp server, stop out bound traffic to www.wasteoftime.com and others. The simulator reads an XML formatted list of possible configuration changes during initialization.

After the students are given sufficient time at the beginning, the instructor can permit interaction between the students. Students may take actions against other students. A list of the possible actions is read from an XML formatted file during initialization. Some of the actions are benign, such as reading the home page of another student's simulated web server. If the other student has their firewall configured too restrictively and prevents this benign action, then the student initiating the action earns a point and the student with the poorly configured firewall loses a point. Students may initiate attacks against the simulated network of another student. If the attack is successful, the attacking student earns a point and the student with the improperly configured firewall loses a point. Students can always see the current scores of all students in the simulation as can be seen in figure 7. The simulation proceeds until the instructor decides to end the session. A "speedometer" is provided for the instructor so they can see the level of interaction between students.

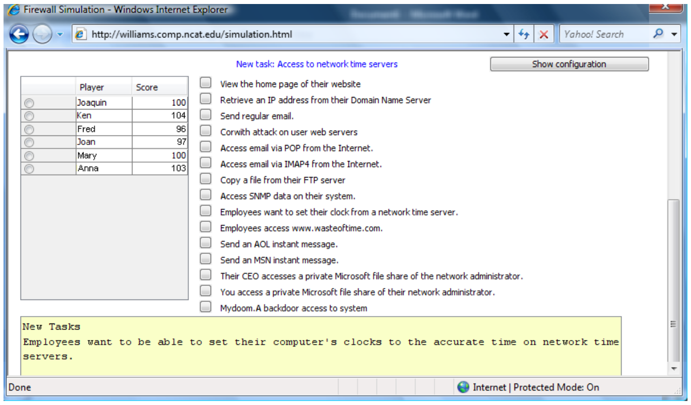
The simulator consists of Java applets running in a web browser. Students simply open the web page of the simulator and can start participating. Additional web pages detail the virtual network while another page provides documentation on configuring firewalls and how to use the simulator. A Java application on the server evaluates the students' firewall configurations and the interaction between students. There is an instructor web page with a Java applet that controls the simulation. The instructor can start and stop a simulation session, add new configuration requirements and send messages to the students.



```
! Create your firewall configuration here.
access-list 123 permit tcp any host 152.8.1.1 eq 80
access-list 123 permit tcp any host 152.8.1.1 eq 443
access-list 125 permit tcp any host 152.8.4.4 eq 23
access-list 130 permit tcp any any eq 123
```

Firewall configuration has been updated

Figure 6 Firewall Configuration Window



Firewall Simulation - Windows Internet Explorer
http://williams.comp.ncat.edu/simulation.html
Yahoo! Search
Show configuration
Done
Internet | Protected Mode: On

New task: Access to network time servers

	Player	Score
○	Joaquin	100
○	Ken	104
○	Fred	96
○	Joan	97
○	Mary	100
○	Anna	103

- □ View the home page of their website
- □ Retrieve an IP address from their Domain Name Server
- □ Send regular email.
- □ Corwith attack on user web servers
- □ Access email via POP from the Internet.
- □ Access email via IMAP4 from the Internet.
- □ Copy a file from their FTP server
- □ Access SNMP data on their system.
- □ Employees want to set their clock from a network time server.
- □ Employees access www.wasteoftime.com.
- □ Send an AOL instant message.
- □ Send an MSN instant message.
- □ Their CEO accesses a private Microsoft file share of the network administrator.
- □ You access a private Microsoft file share of their network administrator.
- □ Mydoom.A backdoor access to system

```
New Tasks
Employees want to be able to set their computer's clocks to the accurate time on network time
servers.
```

Figure 7 Student Firewall Simulator Interface

4. LAN Attack on a Switched Network Simulator

The LAN attack on a switched network simulator is designed to help students understand the methods that attackers use to become Man-In-The -Middle on a switched network. Although, several tools exist that can do Man-In-The -Middle, the technical details are hidden from the users [2, 5]. For example, "Cain & Abel" implements APR (ARP Poison Routing) which enables Man-In-The -Middle attacks to be carried out easily on switched network. However, students cannot learn the details of becoming Man-In-The – Middle which include poisoning the router's ARP table, poisoning the victim's ARP table and forwarding the packets between the router and the victim.

Two tools with which students are asked to carry out a successful Man-In-The –Middle attack on a switched network have been developed. The first tool, named "sendarp", is programmed to send an arbitrary ARP reply message. Students are asked to provide such information as MAC addresses of the source and the target in ARP reply message, IP addresses of the source and the target in ARP reply message, and destination and source MAC addresses in the Ethernet frame header. Only with clear understandings of Ethernet frames, ARP message, and goals of ARP poisoning, can students successfully carry out this attack. This program needs to be executed twice to poison both the router and the victim. To increase the chance of success, the program repeatedly sends the ARP reply message with fixed time interval.

The second tool, named "mim", forwards the packets between the router and the victim. To become a successful Man-In-The –Middle without being detected by the victim, the attacker must forward the intercepted packet either to the victim or to the router and let victim continue communicating without interruption. This tool dumps the intercepted traffic into a file in tcpdump format which can later be viewed using a Wireshark [10] or other packet analyzers. This program asks students under which condition a packet should be forwarded to the router or the victim. Students need to know the format of the IP packet intercepted by the attacker to correctly forward the packets. To assist the lab, we developed a shell script that runs on the victim and constantly contacts a web server that computes a simple mathematical function on the random number sent by the victim. The students need to intercept enough traffic to successfully guess the function computed by the web server.

5. Workshop and Assessment

The developed interactive software simulation education tools have been demonstrated in the 2008 Faculty Development Workshop on Cyber Games and Interactive Simulation. The workshop took place from June 22 to June 28 at University of North Carolina at Charlotte. Twenty-seven faculty members applied to attend the workshop. Because of the grant limitation eighteen faculty members, which came from different universities and colleges, were selected that includes two faculty members from HBCU and four female faculty members.

We presented these interactive education simulators to attendees and let them get hands-on experience. Each of the attendee used one computer that provided required environment and all interactive simulation tools. For each of these education tools we gave a short introduction, explained how to run it, then let attendees use the tool as students will do. The attendees started the tool, followed instructions step by step, answered the challenge questions and run the tools. We also distributed LAN attack tools to the workshop attendees and let them capture the traffic of the victim. About 20% of them were able to complete the assignment independently within one and half hours. Most of the attendees were able to finish the assignment with help of varying degrees. Many attendees thought it was a challenging lab.

At the end of the workshop we conducted a survey. We listed each individual tool to let attendees to rank it and comment on it, therefore, we can improve these tools based on attendees' feedback. The assessment result is shown in table 1. The results show attendees are very interested in these education tools and have very positive experiences using them. Some of the attendees suggested improving the user interface. At the end of the workshop the attendees received all source codes and user manuals and will integrate these tools into their curriculum to allow their students to get hands-on experience.

Table1: Workshop Attendees Survey Results

Survey Questions	**Response**
Please rate the workshop in terms of quality:	
Animation Tools for IA Concepts:	
Packet Sniffer Simulator	78% excellent 11% good 11% gave suggestion
An Animated Learning Tool for Kerberos Authentication Architecture	72 excellent 16% good 12% gave suggestion
A Visualization Tool for Wireless Network Attack	72 excellent 16% good 12% gave suggestion
Interactive SYN Flood Simulator	72 excellent 16% good 12% gave suggestion
Firewall:	
Firewall Configuration and Function	84 excellent 16% good
LAN Attack:	
ARP Spoofing	67 excellent 11% good 11% average 11% gave suggestion
Port Stealing	61 excellent 16% good 11% average 11% gave suggestion

6. Conclusion

We have developed four interactive simulation tools to aid teaching the concepts of packet sniffer, Kerberos authentication architecture, wireless network attacks and SYN flood. We have also developed firewall configuration and function to help students understand how firewall works and what functions it can provide. Two LAN attack laboratories have been implemented to help students strengthen the knowledge of Man-In-The –Middle attack by hands-on experimentation.

In order to improve the instructional capability of IA faculty we demonstrated and distributed these tools to the attendees of the 2008 Faculty Development Workshop on Cyber Games and Interactive Simulation. We explained how these tools work and how we integrated them to our IA curriculum. At the end of the workshop we conducted a survey. The result of the attendees' survey is excellent. We also distributed all source codes and user manuals the attendees. All attendees will integrate these education tools to their IA curriculum during academia 2008-2009. Our experience exhibits by using these interactive education tools and labs the students get hands-on experience and get a deep understanding of the concept of information assurance.

More broadly, these tools can been used in computer science, information management system, computer engineering in any colleges and universities to help students understand IA concepts and get hands-on exercises as part of IA curricular activities.

Acknowledgment

The authors would like to thank National Science Foundation under the award number DUE-0516415 and DUE-0723491 for their support to develop these education tools and conducted the workshop.

References

[1] Baxley, T., Xu, J., Yu, H., Yuan, X., Brickhouse, "LAN Attacker: A Visual Education Tool", Proceedings of 2006 Information Security Curriculum Development Conference, September 2006.

[2] Cain & Abel, http://www.oxid.it/cain.html

[3] Conklin, A., "Cyber Defense Competitions and Information Security Education: An Active Learning Solution for a Capstone Course", Proceedings of the 39th Annual Hawaii International Conference on System Sciences, January 2006.

[4] CERT® Advisory CA-1996-21 TCP SYN flooding and IP Spoofing Attacks http://www.cert.org/advisories/CA-1996-21.html

[5] Ettercap, http://ettercap.sourceforge.net/

[6] Internet Security Systems http://www.iss.net/security_center/advice/Exploits/TCP/SYN_flood/default.htm

[7] Liboon, D., Xu, J., Yu, H., Zhang, J., Yuan, X. and Chu, B., "An Educational Visualization Tool for DDoS Attack", Proceedings of the First Annual Conference on Education in Information Security, October 2006.

[8] Tibbs, Richard and Oakes, Edward, "Firewalls and VPNs: Principles and Practices", Prentice Hall, 2006

[9] Vigna, G., "Teaching Hands-on Network Security: Testbeds and Live Exercises", Journal of Information Warfare, Vol.3, No.2, 2003.

[10] Wireshark, http://www.wireshark.org/

[11] Yuan, X., Vega, P., Xu, J., Yu, H. and Li, Y., "Using Packet Sniffer Simulator in the Class: Experience and Evaluation", Proceedings of ACM Southeast Conference, March 2007.

[12] Yuan, X., Qadah, Y. Xu, J., Yu, H. and Archer, R. "An Animated Learning Tool for Kerberos Authentication Architecture", Journal of Computing Sciences in Colleges, Vol. 22, No. 6 2007.

[13] Yuan, X. Archer, R. L., Xu, J. and Yu, H. "A visualization tool for wireless network attacks", Proceedings of EISTA 2008 - the 6th International Conference on Education and Information Systems, Technologies and Applications, June 29-July 2, 2008.