

A Visualization Tool for Wireless Network Attacks

Xiaohong YUAN, Ricky L. ARCHER, Jinsheng XU, Huiming YU
Department of Computer Science, North Carolina A&T State University
Greensboro, NC 27411, USA

ABSTRACT

The demand for computer security professionals has caused the increase of information security curricula in the universities. Information security courses benefit from such educational resources as hands-on activities, laboratory experiments and concept visualization. This paper presents a visualization tool that demonstrates various attacks popular in wireless networks. This visualization tool is intended to be used in an undergraduate level computer security course or a computer network course. The tool has been used and evaluated in two information security related classes in the Fall 2007 semester and has demonstrated effectiveness in assisting student learning.

Keywords: wireless network attacks, visualization tools, information security education, Evil Twin, Man in the Middle, ARP Cache Poisoning, and ARP Request Replay

1. INTRODUCTION

Due to the increased demand for computer security professionals, there has been a significant increase in the number of security courses in computer science curricula [1-4]. Such courses have benefited from such educational resources as hands-on activities, laboratory experiments and student competitions [5, 6]. Concept visualization, or pedagogical visualization, has been used in computer science education in the fields of algorithm, computer networks, computer architecture, and operating systems [7, 8, 9, 10]. It can greatly benefit information security education as well.

The topic of computer network attacks is an important component in information security curriculum. In this paper, a visualization tool for wireless network attacks is presented. This tool animates the following attacks popular in a wireless networks: Eavesdropping, Evil Twin, Man in the Middle, ARP Cache Poisoning, and ARP Request Replay. This visualization tool is intended to be used in an undergraduate level computer security course or a computer network course. It can be used as classroom instructor demo, classroom student exercises, web-based student learning resources or web-based student assignments.

This work is part of our effort of developing a series of visualization tools for teaching computer security. Other

tools we have developed include: 1) the packet sniffer simulator [11] which demonstrates the packet sniffing vulnerability and the related networking concepts; 2) “LAN Attacker” [12] which demonstrates network attacks on local area networks, such as ARP Poisoning, Port Stealing and MAC Flooding; 3) Visualization of Distributed Denial of Service (DDoS) [13]; and 4) An animated learning tool for Kerberos Authentication Architecture [14] which demonstrates the design of Kerberos. Different from these tools, the visualization tool for wireless network attacks focus on attacks in the wireless environment, though the eavesdropping concept is similar to the packet sniffer, and the ARP cache poisoning attack is similar to that in “LAN Attacker”.

Other work on security concept visualization includes the visualization of buffer overflow attacks [15], cryptography visualizations which animate the process of encryption/decryption [16, 17], and security protocol visualization which demonstrates visually arbitrary protocols in a user-controlled stepwise manner [18].

The remainder of this paper is structured as follows. Section 2 introduces various attacks popular in wireless networks. Section 3 describes the visualization tool. Section 4 discusses the evaluation results of the tool, and Section 5 concludes the paper.

2. WIRELESS NETWORK ATTACKS

The visualization tool we developed demonstrates the following basic attacks: Eavesdropping, Evil Twin, Man in the Middle, ARP Poisoning, and ARP Request Replay. These attacks are described below:

Eavesdropping

In an eavesdropping attack, the attacker configures his/her network interface into promiscuous mode, which allows a network device to read each network packet that arrives at the device. In a wireless environment, any directional antenna can detect IEEE 802.11 transmissions under the right conditions from miles away. Once the attacker has access to the transmission, he/she can then capture packets. Sensitive data such as passwords and other credentials are now readily available to the hacker if they are not encrypted. There are many easily available applications that can be used for eavesdropping, including *bsd-airtools* [19] and *Kismet* [20].

Evil Twin

An evil twin is a wireless access point (AP) that masquerades as a legitimate one. An attacker can create an evil twin by simply using a laptop, a wireless card and some readily-available software. The attacker positions himself in the vicinity of a legitimate AP and discovers what service set identifier (SSID) and radio frequency the legitimate AP uses. He then sends out radio signals using the same SSID. The attacker's computer becomes a Rogue AP or evil twin. Since the Rogue AP may be physically closer to the user than the legitimate AP, the user may be connected to the evil twin rather than the legitimate AP. A tool such as WifiBSD [21] and Aircsnarf [22] can be used to set up an evil twin.

Man in the Middle

In the Man in the Middle (MITM) attack, the attacker intercepts the traffic between two computers. The attacker sniffs packets from the network, may modify the packets and inserts them back into the network. In wireless environment, the attacker can set up a Rogue AP. When the user associates with the Rogue AP, the Rogue AP can be the man in the middle between the user and the legitimate AP.

ARP Cache Poisoning

Address Resolution Protocol (ARP) is a network layer protocol used to associate an IP address with a MAC address. A network device has an ARP cache, which contains all the IP addresses and MAC addresses the device has already matched together. With the ARP cache the device does not have to repeat ARP Requests for devices it has already communicated with. In an ARP cache poisoning attack, the attacker introduces erroneous IP to MAC address mapping in another host's ARP cache. This results in IP traffic intended for one host being diverted to a different host, or to no host at all.

ARP Request Replay

Replay is a network attack where a validated data transmission is intercepted and retransmitted at a later

time. An ARP request replay attack is used by the attacker to generate new initialization vectors (IVs), which can be used to crack the WEP encryption key. The attacker first eavesdrops an ARP request on the network and then retransmits it back to the network. When the computer with the IP address in the ARP request receives the ARP request, it sends an ARP response to the attacker. Every time the computer sends an ARP response, it generates a new IV which is captured by the attacker. The attacker repeatedly sends out ARP request until large quantities of IVs are captured. These IVs are then used to crack the WEP key. A tool that injects packets for capturing IVs is aircrack-ng [23].

3. THE VISUALIZATION TOOL FOR WIRELESS NETWORK ATTACKS

The visualization 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 user can choose to "play" the animation, or trace the demo step by step. The user can also "rewind" to see the previous step of the animation, "forward" to see the next step of the animation, "pause" or "stop" the animation. The tool also provides "challenge questions" to give the user a quiz on the animation he watched. In what follows, the visualization scenarios of the five types of wireless network attacks are described.

3.1 Eavesdropping

The Eavesdropping scenario demonstrates how a hacker eavesdrops the communication between two wireless nodes. Figure 1 shows a snapshot of the Eavesdropping demo. The network includes two users (Alice, Bob), an AP, and a hacker (John). The messages between two network nodes are represented by a small box. Textual explanation is provided to describe the animation scenario.

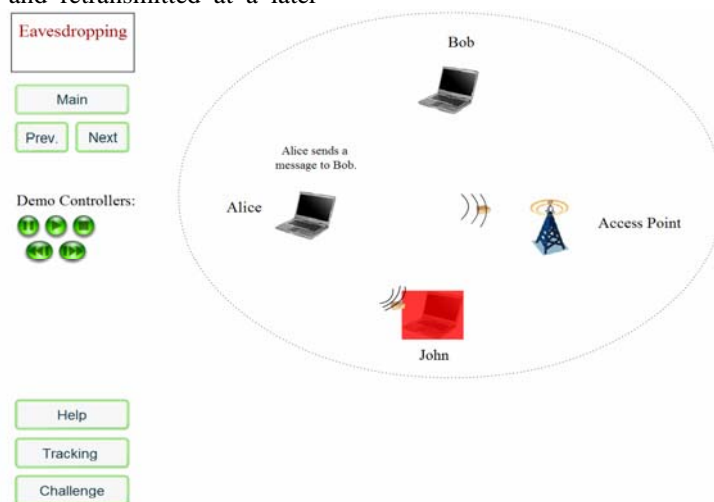


Figure 1. A snapshot of the Eavesdropping demo

The sequence of the animation is described below.

1. John sets his NIC to promiscuous mode. This allows John to capture packets between Alice and the AP.
2. Alice sends a message to Bob. The packets are sent from Alice to the AP, and then forwarded by the AP to Bob.
3. John captures the message since its NIC is configured to promiscuous mode, and its radio is tuned to the communication channel between Alice and the AP.
4. Bob sends a message back to Alice. The packets are forwarded by the AP to Alice.
5. John captures the message (sent by AP to Alice).

3.2 Evil Twin

The Evil Twin scenario demonstrates how the hacker sets up an evil twin access point, and has the client connect to it. The sequence of the animation is described below.

1. John is eavesdropping the communication between Alice and the AP.
2. Alice sends out a PROBE REQUEST requesting the ESSID of an AP nearby. The AP responds with a PROBE RESPONSE which includes its ESSID and BSSID. John captures the ESSID information that is in the PROBE RESPONSE.
3. Alice then sends AUTHENTICATE REQUEST to the AP, and receives AUTHENTICATE RESPONSE from AP; after that, Alice sends ASSOCIATE REQUEST to the AP, and receives ASSOCIATE RESPONSE from the AP. Alice is now associated to the AP whose ESSID is "CORP".
4. John sets up a Rogue AP using the ESSID John captured through eavesdropping.

5. John broadcasts a de-authenticate frame to Alice to disconnect Alice from the AP.
6. Alice is disconnected from the AP.
7. Alice re-associates with the rogue AP since it is physically closer.

3.3 Man in the Middle

The Man in the Middle scenario demonstrates how an attacker sets up a rogue AP and intercepts message between a user and an AP. The sequence of the animation is described below.

1. Alice is connected to the AP. John is eavesdropping and captures the PROBE RESPONSE sent by the AP.
2. John sets up a Rogue AP.
3. John broadcasts a de-authenticate frame to Alice and Alice is disconnected from the AP.
4. Alice re-authenticates to John by accident, or because John is physically closer to Alice.
5. John authenticates to AP on behalf of Alice.
6. The AP responds to John's Re-authenticate request.
7. John responds to Alice's Re-authenticate request.
8. Alice re-associates to John.
9. John re-associates to the AP on behalf of Alice.

3.4 ARP Cache Poisoning

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 2 shows a snapshot of the ARP Cache Poisoning demo. In Figure 2, 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.

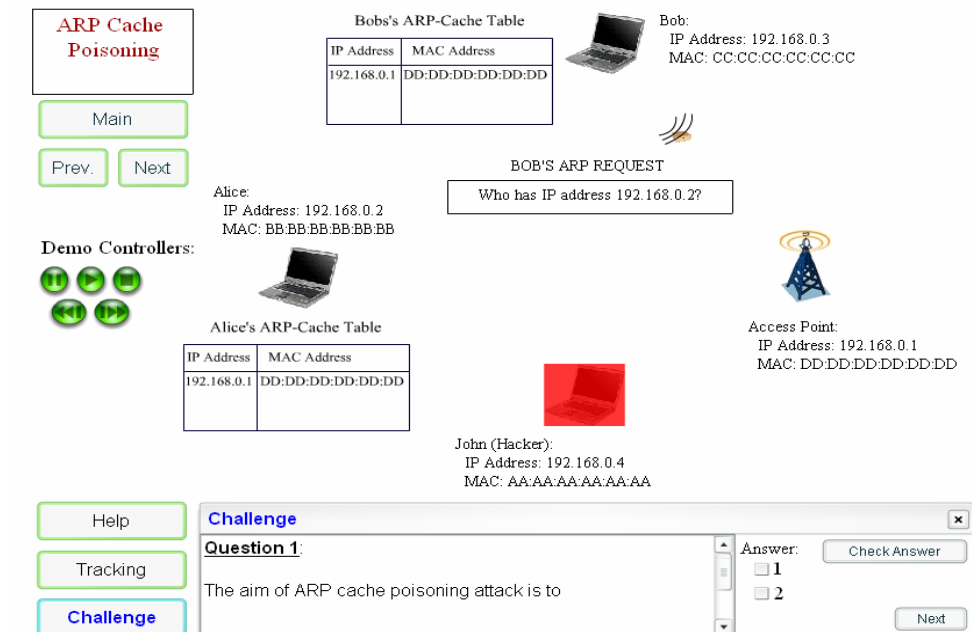


Figure 2. A snapshot of the ARP Cache Poisoning demo

The sequence of the animation is described below.

1. Bob broadcasts an ARP request: "Who has IP address 192.168.0.2"?
2. The ARP request is broadcasted by the AP to Alice and John.
3. Alice sends out ARP response: "I have 192.168.0.2, my MAC address is: BB:BB:BB:BB:BB:BB."
4. Alice's IP and MAC addresses are added to Bob's ARP cache table.
5. John sends to Alice a fake ARP response: I have 192.168.0.3, my MAC address is AA:AA:AA:AA:AA:AA. This message is forwarded by the AP to Alice.
6. Alice's ARP cache table adds the entry 192.168.0.3, AA:AA:AA:AA:AA:AA.
7. Alice sends a packet to Bob; because of the incorrect IP and MAC address mapping, the package is sent to John instead of Bob.
8. John forwards the packet to Bob. John acts as a man in the middle between Alice and Bob.

2. The ARP request is sent to the AP, and is broadcasted by the AP to Alice and John.
3. Alice sends the encrypted ARP response to Bob. The encrypted ARP response includes an IV.
4. John captures the ARP response through eavesdropping, extracts the IV from the packet, and stores the IV in the captured IV table.
5. John resends the captured ARP request: "Who has IP address 192.168.0.2"?
6. The ARP request is broadcasted by the AP to Alice and Bob.
7. Alice sends the encrypted ARP response to John. This ARP response uses a new IV.
8. John receives the response packet, extracts the new IV, and adds it to its IV table.
9. Repeat steps 5 – 8 until John collects enough different IVs for cracking the WEP key.
10. John cracks the WEP key using the captured IVs.

3.5 ARP Request Replay

The ARP Request Replay scenario demonstrates how an attacker replays ARP request in order to crack WEP key. Figure 3 shows a snapshot of the ARP Cache Poisoning demo. In Figure 3, 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. An IV is a continuously changing number used in combination with a secret key to encrypt data.

The sequence of the animation for the ARP request replay scenario is described below.

1. Bob broadcasts an ARP request: "Who has IP address 192.168.0.2"? John captures Bob's ARP request.

4. TOOL EVALUATION

The visualization tool for wireless network attacks was used in two courses in the Fall 2007 semester in the Department of Computer Science, North Carolina A&T State University. The two classes were: "Network Security" and "Security Management for Information Systems". The students in the "Network Security" class are mostly graduate students, and the students in the "Security Management for Information Systems" are all undergraduate students. The students in both classes were given a link to the web site of the tool, and were asked to complete a pre-test quiz before they used the tool and a post-test quiz after they used the tool, and also fill out a questionnaire. This was given as a homework assignment. All together 15 students turned in the homework.

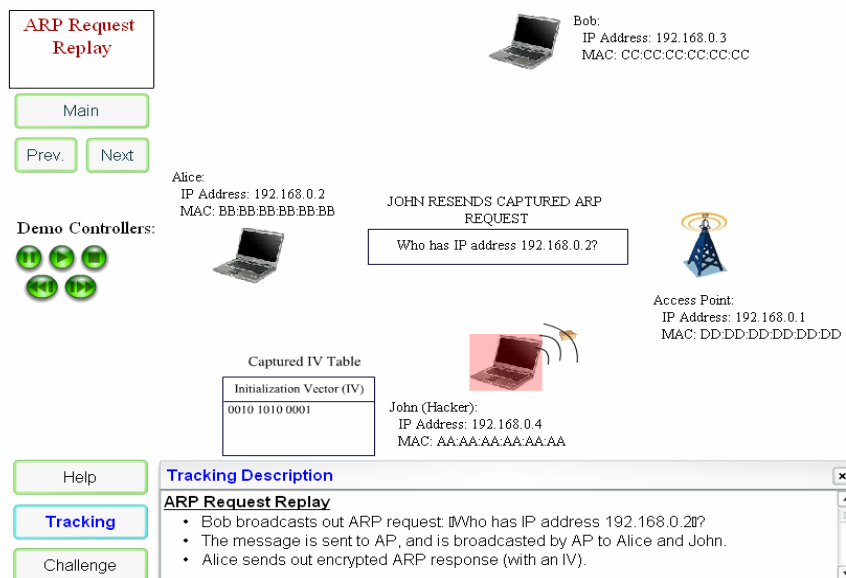


Figure 3. A snapshot of the ARP Request Replay demo

Figure 4 shows the comparison between the pre and post test scores. The pre-test and post-test results show that every student improved from pre-test to post-test.

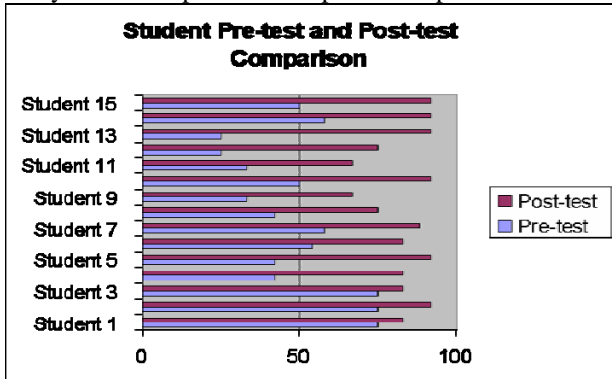


Figure 4. Student pre-test and post-test comparison

Table 1 lists the questions asked in the questionnaire and the student responses to them. The total number of students who participated in the survey is 12. From the student survey result, we can conclude that, overall, the tool has been effective in helping students learn wireless

network attacks. The students had very positive experiences using this tool.

5. CONCLUSION

This paper presents a visualization tool that animates five basic wireless network attacks: Eavesdropping, Evil Twin, Man in the Middle, ARP Cache Poisoning, and ARP Request Replay. This tool can be used by instructors and students in teaching and learning information security. This tool was evaluated by a group of Computer Science undergraduate and graduate students who are currently learning about the topics described in this tool. Pre-test, post-test, and questionnaire are used to evaluate the tool. The evaluation results show that the tool is effective in improving student learning. Future work would include adding more wireless network attack scenarios to the tool framework, adding audio to the animation of each scenario, and conducting more extensive evaluation of the tool.

Table 1. Student survey results

Question	Response
1. Do you enjoy using the tool?	25% strongly agree 75% agree
2. Do you think the tool is easy to use?	67% strongly agree 33% agree
3. Do you feel you understand the concept better when using the tool?	42% strongly agreed 50% agree 8% neither agreed or disagreed
4. How likely are you to recommend this tool to others?	58% definitely will recommend 33% probably will recommend 9% not sure
5. Would you like to see more of these demos (or simulators) in your courses?	92% strongly agree 8% agree
6. How much time did you spend with the tool on average?	The average time of use was 21 minutes
7. What were the 3 things you liked best about this demo?	Easy to use, visual effects, easy to understand, interactive, neat, details, challenge questions, structure
8. What were the 3 things you liked least about this demo?	No sound, not enough detail
9. What would you suggestion to make this tool a better application?	Add sound, more color, make sure the questions increase in difficulty

REFERENCES

- [1] Frincke, D. and Bishop M. Joining the security education community, *IEEE Security and Privacy*, Vol. 2, Issue 5, 2004, pg. 61-63.
- [2] LeBlanc, C. and Stiller E. Teaching computer security at a small college. *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education*, Norfolk, Virginia, USA, March 3-7, 2004, pg. 407-411.
- [3] Mullins, P. et. al. Panel on integrating security concepts into existing computer courses, *Proceedings of the 33th SIGCSE Technical Symposium on Computer Science Education*, Northern Kentucky, Cincinnati, USA, February 27-March 3, 2002, pg. 365-366.
- [4] Bhagyavati, et. al. Teaching hands-on computer and information systems security despite limited resources, *Proceedings of the 36th SIGCSE Technical Symposium*

- on *Computer Science Education*, St. Louis, Missouri, USA, February 23-27, 2005, pg. 325-326.
- [5] Brustoloni, J. C. Laboratory experiments for network security instruction, *Journal on Educational Resources in Computing*, Vol. 6, No. 4, December 2006.
- [6] Haynes, A. and Stratton, T. Cyber defense 2003 & information assurance education, *2003 IEEE International Conference on Systems, Man & Cybernetics*, Oct. 2003.
- [7] Harrold, M. J. and Stasko, J. Algorithm animation, 2002. Available at: <http://www.cc.gatech.edu/gvu/softviz/algoanim/>
- [8] Holliday, M. A. Animation of computer networking concepts, *ACM Journal of Educational Resources in Computing*, Vol. 3, No. 2, June 2003, Article 2.
- [9] Null, L. and Rao, K. CAMERA: Introducing memory concepts via visualization, *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education*, St. Louis, Missouri, USA, February 23-27, 2005, pg. 96-100.
- [10] Carr, S., Mayo, J. and Shene, C.K. ThreadMentor – A system for teaching multithreaded programming, 2003. Available at: <http://www.cs.mtu.edu/~shene/NSF-3/>
- [11] Yuan, X., Vega, P., Xu, J., Yu, H., and Providence, S., An animated simulator for packet sniffer”, *Proceedings of WECS7 - The Seventh Workshop on Education in Computer Security*, January 4-7, 2006, Monterey, California.
- [12] Baxley, T. et. al. LAN Attacker: A visual education tool”, *Proceedings of the 2006 Information Security Curriculum Development Conference*, pp. 137-142, September, 2006.
- [13] Liboon, D. et. al. An educational visualization tool for DDoS attack, *Proceedings of the ACEIS 2006 – First Annual Conference on Education in Information Security*, pp. 48-53, September, 2006.
- [14] Yuan, X, Qadah, Y., Xu, J., Yu, H., Archer, R., and Chu, B. An animated learning tool for kerberos authentication architecture, *Journal of Computing Sciences in Colleges*, Vol. 22, No. 6, 2007.
- [15] Crandall, J.R., et. al. Driving home the buffer overflow problem: a training module for programmers and managers, *Proceedings of National Colloquium for Information Systems Security Education (NCISSE 2002)*, Washington, 2002.
- [16] Gerhart, S. et. al. Increasing security in aviation-oriented computing education: a modular approach, August 2005. Available at: <http://nsfsecurity.pr.erau.edu/>
- [17] Schweitzer, D. and Baird, L. The design and use of interactive visualization applets for teaching ciphers, *Proceedings of the 2006 IEEE Workshop on Information Assurance*, 2006, pp. 69-75.
- [18] Schweitzer, D., Baird, L., Collins, M., Brown, W., and Sherman, M. GRASP: A visualization tool for teaching security protocols, *Proceedings of the 10th Colloquium for Information Systems Security Education*, June, 2006. pp. 75 – 81.
- [19] The FreeBSD Project. Available at <http://www.freebsd.org>, accessed on December 18, 2007.
- [20] Kismet. Available at <http://www.kismetwireless.net>, accessed on December 18, 2007.
- [21] WifiBSD. Available at <http://www.wifibsd.org>, accessed on December 18, 2007.
- [22] Airsnarf. Availabe at <http://airsnarf.shmoo.com>, accessed on December 18, 2007.
- [23] Aircrack-ng. Availabe at <http://www.aircrack-ng.org/doku.php?id>, accessed on December 18, 2007.