In Proceedings USENIX Workshop on Cyber Security and Test, July 2008

SWOON: A Testbed for Secure Wireless Overlay Networks

Y. L. Huang[†], J. D. Tygar[*], H. Y. Lin[‡], L. Y. Yeh[‡], H. Y. Tsai[†], K. Sklower[*], S. P. Shieh[‡], C. C. Wu[‡], P. H. Lu[†], S. Y. Chien[‡], Z. S. Lin[†], L. W. Hsu[‡], C. W. Hsu[‡], C. T. Hsu[†], Y. C. Wu[‡], M. S. Leong[*]

[†]*Department of Electrical and Control Engineering, National Chiao-Tung University, Taiwan*

[*]*Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, USA*

[‡]*Department of Computer Science, National Chiao-Tung University, Taiwan*

Abstract

There is strong demand for solutions to security problems in various wireless networks, such as WiFi, WiMAX, 3GPP and WSN, not only for the individual networks themselves but also for the integration of these networks. A complete solution cannot be proposed by piecemeal proposals but requires a holistic examination of all security concerns. The solution requires assessment tools, such as wireless testbeds for designing and testing wireless security technologies. We describe a comprehensive and flexible wireless testbed allowing designers to test their systems without actually building a physical test environment. Moreover, such a testbed can also shorten the test cycle and the time to market. Our SWOON testbed uses two experimental nodes to simulate one single wireless node. Such a pairing design helps reduce the porting efforts of wireless drivers and thus increase the flexibility for adapting various wireless interfaces in the SWOON testbed. We verify the feasibility and stability of the SWOON testbed by conducting distributed denial of service (DDoS) and eavesdropping experiments. In the future, the SWOON testbed will be extended to support heterogeneous wireless networks, such as WSN, WiMAX or 3GPP.

1 Introduction

Cyber-security problems need special attention in wireless networks, such as 802.11 a/b/g, 802.16 d/e, etc. A wide variety of research has addressed the cyber-defense of wireless networks. However, such research has been limited by the lack of a secure, configurable experimental infrastructure for reproducible experiments validating new designs and new technologies in realistic scenarios. Sometimes it is also insufficient to use network simulators directly since these simulators, abstracting some system attributes, do not model the bottlenecks from experimental nodes, such as CPUs, buses, devices and drivers.

The SWOON (Secure Wireless Overlay Observation Network) testbed is an emulation-based testbed for real world experiences and scalable tests over an overlay network, consisting of wireless sensor networks, 802.11 a/b/g, etc. It can evaluate protocols, mechanisms and techniques for secure wireless communication. Researchers and designers can create their own topologies and run experiments on the SWOON testbed without re-establishing and re-installing hardware and software modules required for their wireless networks. In addition, the SWOON testbed also allows researchers to monitor the network traffic, evaluate the performance of the protocols under test and validate the researches they presented.

SWOON is developed on top of the Defense Technology Experimental Research (DETER) testbed [1]. DETER provides an experiment platform for investigating security issues. Based on Emulab [2], DETER offers an experimental infrastructure with safe and repeatable configurations. The current scale of DETER has been increased to hundreds of nodes, meaning that it is capable of dealing with medium-scale security experiments, such as DDoS and worm behavior experiments running in EMIST project [3]. The DETER project provides a safe testbed that can match the threat level of the experiments. DETER in its current form addresses only *wired* networks. SWOON builds on DETER adding support for *wireless* networks.

Two DETER nodes are used to simulate a single wireless node in SWOON. One node serves as the *application node* running various applications, while the other, the *shadow node* emulates aspects of the wireless network interface, delivering packets for its application node. These two nodes form an *application-shadow node pair*. Researchers can simulate the behavior of a wireless interface, such as delay, loss and jitter, on the shadow node. Using mechanisms implemented in DETER, researchers can create and run experiments on the SWOON without interfering with each other.

SWOON features a friendly graphical user interface (GUI) allowing researchers to easily set up the desired topology and parameters, such as network type, radio coverage, bandwidth, delay, loss, etc. Researchers can monitor the experiment results through the GUI. SWOON provides a platform to emulate various attacks, including but not limited to, unauthorized access, spoofing, denial of service, flooding, man-in-the middle, drive-by spamming, wireless eavesdropping and DDoS attacks.

This paper details the design and development of SWOON. We briefly introduce existing testbeds in Section 2. Section 3 illustrates the design of the SWOON testbed. Section 4 and 5 show the design of user interface and the experiments for wireless security, respectively. We compare testbeds and give open problems in Section 6 and 7.

2 Related Work

In this section, we briefly introduce existing testbeds: Emulab, DETER, ORBIT radio grid testbed and Agarwal's wireless emulator.

2.1 Emulab

Emulab [2], developed by the University of Utah, is an emulation platform for research in distributed systems and networks. In Emulab, a set of experimental nodes is flexibly connected in a network topology described using the NS (Network Simulator) language. Experiments are isolated by programming various VLANs (Virtual Local Area Networks) in Emulab. Nodes configured in the same VLAN can communicate as if they were attached to the same wire, regardless of their physical location. Thus, Emulab can run multiple experiments simultaneously and guarantees no interference between experiments. To set up a new experiment, Emulab maps the desired network topology to the physical network by taking the following steps: 1) allocates the experimental nodes and switches; 2) configures the VLANs to construct the desired topology, and 3) loads the designated executable images to the specified experimental nodes, so users are able to perform real world testing.

2.2 DETER

Built using Emulab, the DETER testbed [1] provides infrastructure for conducting repeatable experiments in computer security, especially those involving malicious code. DETER testbed allows remote access for experiments while keeping the experiments themselves contained within the testbed. Since it is intended to support security-related experiments [4][5][6], containment and security are the basic requirements in designing such a testbed. The design of the DETER testbed is an effective compromise of the goals of experimental fidelity, repeatability, programmability and research functionality. In addition, motivated to enlarge the scale of an experimental infrastructure for realistic wired network testing, the DETER testbed is composed of two interconnected Emulab testbeds. IPSec tunnels connect the experiment and control switches, respectively. Firewalls provide a container to isolate experiments from public networks. This is critical since some experiments may contain viruses or conduct attacks that may threaten the outside network.

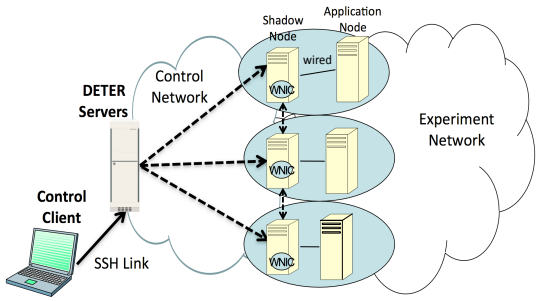

Figure 1: SWOON Architecture. The control client commands the application-shadow node pairs through the DETER servers.

2.3 ORBIT and Agarwal's Wireless Emulator

ORBIT [7] is a two-tier wireless testbed designed for 3G and 802.11 networks. The testbed comprises a grid of 802.11 nodes and can dynamically interconnect these nodes into specified topologies. Each ORBIT radio node is a real device (PC) with two Ethernet ports and two 802.11 interfaces, rather than an emulated, configurable device as in SWOON.

V. Agarwal [8] implemented a wireless network emulator with emulated 802.11 MAC and PHY layers logically inserted between the IP layer and 802.3 MAC layer. Agarwal did not use shadow nodes as we propose here, so his system requires significant porting efforts in the IP network protocol stacks.

3 SWOON Testbed

This section describes the system architecture of the SWOON testbed, including the interaction with DETER servers, the control client, application-shadow node pairs and secure virtual links to communicate with DETER experimental nodes. Fig. 1 shows the system architecture of the SWOON testbed.

DETER Servers. Two servers, "Boss" and "User", control the experiments in the DETER testbed. The "Boss" server controls the switches and power controllers. It allocates experimental nodes, interconnects them by setting up VLANs in the switches and creates topologies specified by the experimenters. The "User" server manages user accounts for experimenters. Through "User",

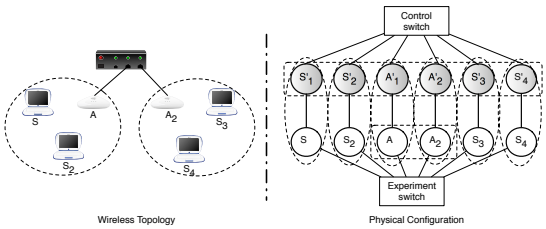

Figure 2: Wireless Network Topology with two APs and four STAs. Six shadow nodes and six application nodes are used to construct the experiment topology. The graph on the left-hand side is the desired wireless topology for experiments. The graph on the right-hand side presents the physical experimental network configured in the DETER testbed.

experimenters can remotely and securely access experimental nodes using relayed SSH as a communication medium.

Control Client. The control client provides a graphical configuration tool for experimenters to specify desired network topologies. Through the control client, experimenters can run specified wireless experiments by sending commands to corresponding shadow nodes. The control client converts specified topologies into configuration files, which initialize wired network topologies on the DETER testbed. The control client also transmits a coverage table tailored for each node. The coverage tables record the distances of neighboring nodes in the transmission range of an application node.

Application-Shadow Node Pair. The application-shadow node pair is a core contribution of SWOON. It is a pair of nodes, an application node and a shadow node. The application node runs applications while the shadow node emulates the wireless interface. Such a pairing design makes our testbed OS-independent. No driver porting or kernel modification is required in either application node or shadow node. Without the need of installing real wireless devices, our design also allows higher flexibility in adapting to various kinds of wireless interfaces.

Application Node: An application node is a regular node that runs applications. The application node connects to its pairing shadow node via an ether link. It sends data packets to its ethernet interface, and those packets are routed to the shadow node. For example, when emulating an 802.11 a/b/g network, an application node may be a station (STA) or an access point (AP).

Each STA in an emulated 802.11 network requires an interface to associate and communicate with an AP. In the SWOON testbed, the application node emulating the STA application requires at least one interface for its shadow node. The interface routes the application data packets to the corresponding shadow node and emulates communication with the AP via its shadow node.

Each AP in an emulated 802.11 network requires at least two interfaces: 1) an ethernet interface connected to local network and 2) a wireless interface serving STAs within its coverage. In the SWOON testbed, the application node presenting an AP, requires at least two network interfaces, one connected to the switch and the other to its shadow node. The first interface connected to the switch is in charge of delivering data packets to other nodes in the local wired network. The second interface connected to its shadow node is responsible for broadcasting packets to the shadow nodes of the STAs within the AP's coverage.

Shadow Node: A shadow node acts as a virtual wireless network interface for an application node. It can emulate various replaceable Media Access Control (MAC) layers, such as 802.11 and WiMAX, for its application node. To simulate radio signals on a wired testbed, the shadow node broadcasts the packets to all reachable nodes in the same VLAN. Each shadow node is equipped with two interfaces: one is connected to its application node and the other to the switch. Fig. 2 illustrates an example of constructing wireless networks on the DETER testbed.

In Fig. 2, a wireless network topology with two APs and four STAs is constructed using six shadow nodes and six application nodes. To emulate the broadcast in wireless networks, three types of VLANs are configured in the "experiment switch".

- The six shadow nodes are configured in one VLAN. This emulates the broadcast of wireless packets between the two APs and four STAs.
- The two application nodes, running AP applications, are also configured in one VLAN. This emulates the data link between the two APs which are connected to the same switch.
- The application-shadow node pair is configured in one VLAN. This limits the direct route between the application node and its shadow node.

The shadow node simulates wireless network behavior and emulates MAC layers of wireless technologies. A Wireless Network Interface Card (WNIC) emulator, running on the shadow node, is in charge of the simulation and emulation. Taking 802.11 as an example, the WNIC emulator performs the following operations on packets:

- capture packets sent from its application node by using *pcap*,
- determine whether to delay or drop packets according to the parameters, including delay, loss, jitter and bandwidth, specified by users,
- encapsulate/decapsulate two headers in the packets if not to drop them, and
- broadcast outgoing packets via UDP sockets or forward incoming packets to upper-layer applications.

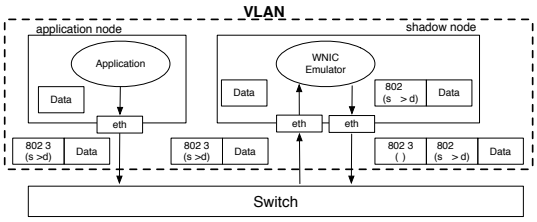

Figure 3: Packet Flow in the Application-Shadow Node Pair: The application data packets are routed to the shadow node. The WNIC emulator of shadow node receives and unpacks the 802.3 header. Then, the data packets are repacked with new 802.11 headers to broadcast to other nodes in the network.

The two headers prepended to the outgoing packets are the 802.11 header (inner) and 802.3 broadcast header (outer). Similarly, upon receiving a broadcast packet from another shadow nodes, the WNIC emulator unpacks the 802.3 broadcast header and 802.11 header. Fig. 3 illustrates the packet flow between the application node and shadow node. The implementation of the WNIC emulator is realized at user space.

The WNIC emulator also simulates wireless network behavior according to the parameters specified by users, including delay, loss, jitter and bandwidth. These can be set to model physical behavior as needed. Then, the emulator determines whether to delay or drop the data. Since the loss rate and latency of a wireless network are related to the signal strength, the distance between each node can be used to determine the rate. The node distance can be calculated by the control client and stored in the coverage table, which is then broadcast to all experimental nodes together with every configured parameter during initialization. The WNIC emulator also performs association, disassociation and authentication for its application node. Therefore, such an implementation can realize the emulation of node mobility.

Virtual Link. The DETER testbed isolates private networks from public networks. This protects the private network from exterior malicious attackers and prevents errant applications in the private network from affecting outside networks. To allow users to safely run reproducible experiments, network attacks and countermeasures, only two external nodes, "Boss" and "User" servers, can access the experimental nodes in DETER. Since the control client is located outside the DETER testbed, we setup a virtual link to offer proxy service on "User" server, as shown in Fig. 1, to relay packets from control client to experimental nodes. The virtual link is implemented using an SSH tunnel to provide a channel to monitor the experiment and the behaviors of the nodes, such as movement, handover, etc.

Such a link can be used to bypass the firewall and connect to the control client via a proxy server. Since there are only node names embedded in the commands

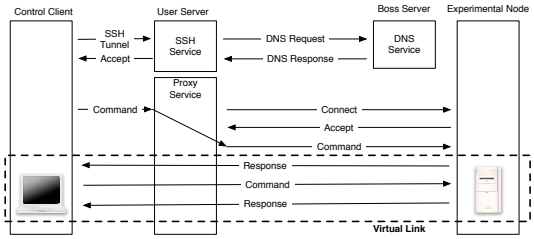

Figure 4: Message Flow for Constructing a Virtual Link: The control client connects to the "User" server using an SSH tunnel and commands experimental nodes through the tunnel.

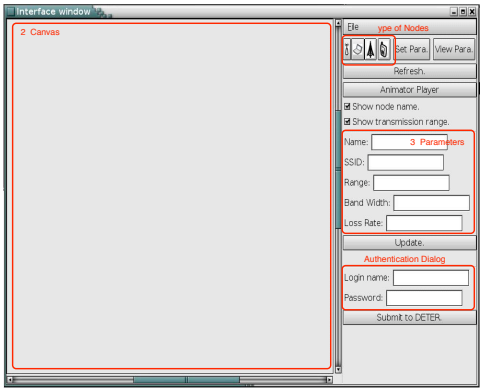

Figure 5: SWOON GUI: Four major components are designed to provide easy setup and real-time experimental results.

sent by the control client, the DNS service running on the "Boss" server is also required to forward these commands. Similarly, the responses are forwarded back to the control client through this proxy server. Fig. 4 shows the message flow for constructing the virtual link.

4 User Interface

The section introduces GUI for the SWOON users. This interface is a graphical tool used for defining, configuring, controlling, loading and monitoring experiments remotely. Fig. 5 shows four major components in designing the SWOON GUI: (a) types of nodes; (b) canvas; (c) parameters; and (d) authentication dialogue for logging into the DETER testbed.

Types of Nodes. There are four types of nodes supported by this version of the GUI: 802.11 STAs, 802.11 APs, WiMAX subscriber stations (SSs) and WiMAX base stations (BSs). The 802.11 STAs connect to the Internet via an 802.11 AP. The WiMAX SSs connect to a WiMAX BS via 802.16d to obtain network resources.

Canvas. The canvas presents a visualization of the wireless topology. Experimenters can place network components on the canvas, and configure the distance between each components. The canvas can also show the transmission coverage of each component. Upon running an experiment, the canvas shows the real-time result of the experiment.

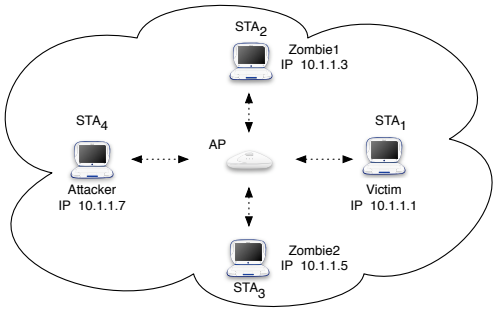

Figure 6: Wireless Network topology for Experiment Setup: DDoS.

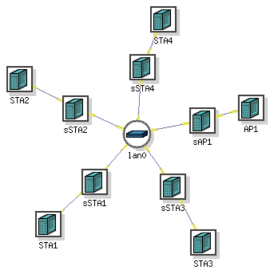

Figure 7: Physical Network Topology on DETER for DDoS Experiment.

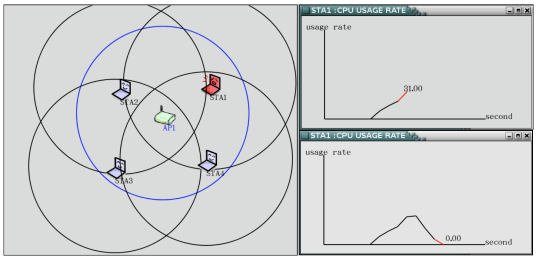

Figure 8: Experiment Result: Under DDoS Attack.

Parameters. Experimenters can set the attributes of each experimental node by using parameters such as coverage range, bandwidth, loss rate, OS, etc. Experimenters can adjust most attributes in real-time during the experiment.

Authentication Dialogue. To securely control the access to the experiments, an authentication dialogue is used to allows user to log in the DETER testbed and swap in the specified experiments.

The SWOON GUI allows experimenters to easily design the desired topology and configure the attributes of each component. The wireless topology specified on the canvas is converted to a configuration file called a NS file. The DETER servers initialize the wired network topology according to the NS file. The SWOON GUI also generates a coverage table based on the topology and transmits the coverage table to the shadow nodes in the experiment.

5 Experiments

Here we describe some exemplar experiments that we have run. The first experiment simulates a DDoS attack and the second one shows the ability of our system to simulate wireless eavesdropping.

5.1 DDoS

DDoS is a common attack on the Internet. It is an attack in which a multitude of compromised or zombie systems attack a single victim. The flood of incoming messages to the victim system forces it to shut down, thereby denying service to legitimate users of the victim system.

To build up a DDoS experiment, we deployed the topology shown in Fig. 6. $n+2$ wireless hosts are associated with a common AP; one of them is the attacker, which unites n zombies to attack the victim. To run such an experiment on SWOON, we need $2n+6$ nodes: $n+2$ STA nodes, one AP node and $n+3$ shadow nodes. Since all STA nodes communicate with a single AP, all the shadow nodes can be configured in one VLAN to physically broadcast packets. Each application node and corresponding shadow node are configured in the same VLAN, so packets from application nodes are directly routed to the paired shadow nodes. Fig. 7 shows the topology configured for DDoS attack in DETER.

We use Tribe Flood Network 2000 (TFN2K) [9] for this experiment. The tool is installed on all the attacking nodes, including the attacker and its zombie hosts. For the victim host, we use Redhat Linux 9, the legacy release, whose network stack implementation is vulnerable to DDoS attack. With Simple Network Management Protocol (SNMP), we can observe and monitor the packet receiving rate and CPU utilization on SWOON GUI. When starting the attack, the packet receiving rate of the victim node rises rapidly. The victim node then detects the abnormal receiving rate and issues an alert. After stopping the attack, the packet receiving rate in the victim node returns to normal state, as shown in Fig. 8.

This experiment is not only of interest in its own right, but also demonstrates the stability of SWOON under stress.

5.2 Wireless Eavesdropping

Compared with wired networks, wireless networks are more vulnerable to threats and attacks by intruders. Since packets are broadcast to other nodes, it is easy for a node to eavesdrop on packets transmitted over the wireless network. For the wireless eavesdropping experiment, two nodes are communicating with each other via the telnet protocol. Fig. 9 shows that the third node eavesdrops on the conversation.

In the eavesdropping experiment, each wireless node in SWOON is represented by an application-shadow node pair in DETER. This experiment requires eight DETER nodes: one AP node, three STA nodes and four

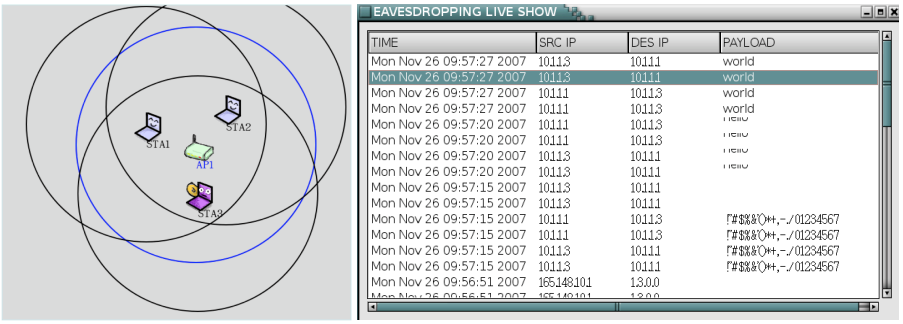

Figure 9: Experiment: Wireless Eavesdropping Attack.

Table 1: Emulable Attacks

Attacks/Testbeds	DETER	SWOON
War Driving	N	Y
MAC Spoofing	N	Y
IP Spoofing	Y	Y
Eavesdropping	Y	Y
Wireless Eavesdropping	N	Y
Man-in-the-Middle	Y	Y
Evil Twin	N	Y
DDoS	Y	Y

shadow nodes. The experiment shows how effective is SWOON in emulating wireless monitoring.

6 Comparing SWOON and DETER

Table 1 shows attacks that can be emulated on SWOON and a vanilla version of DETER. War driving, for example, requires emulation of a mobile agent that searches for wireless networks and collects data packets. DETER is not suited to emulate mobile agents. SWOON can emulate mobile agents, and can have the emulated agents run war driving programs such as NetStumbler [10] or SWScanner [11]. Experimenters can run protocols in application nodes and verify how robust those protocols are against war driving attacks.

7 Future Work

SWOON is a valuable testbed for studying and observing the security issues in wireless networks. It provides high flexibility in constructing wireless topology and dynamically adjusting parameters. With SWOON, designers can run their experiments without re-installing and re-configuring hardware devices and software modules. The wireless topology and the parameters can be dynamically adjusted through a user-friendly GUI. The designers can get the results of the experiments as soon as the new values are applied. SWOON can emulate mobile wireless devices. By setting roaming paths, the handover of devices can be monitored on the SWOON GUI in real time.

Currently, the SWOON testbed only supports an emulation platform for 802.11 networks. We have began implementing real wireless sensor networks on SWOON in the near future. This allows designers to physically run experiments, such as secure aggregation and broadcasting on these sensor nodes. We are also adding support for other network types, including WiMAX, 3G and ZigBee. SWOON provides wireless network designers with an efficient, cost-effective tool for testing and evaluating protocols and modules.

Acknowledgment

This effort was partially supported by the International Collaboration for Advancing Security Technology (iCAST) and Taiwan Information Security Center (TWISC) projects, sponsored by National Science Council under the grants NSC96-3114-P-001-002-Y and NSC96-2219-E-009-013, NSC-97-2918-I-009-005, respectively.

References

[1] T. Benzel *et al.*, "Experience with DETER: A Testbed for Security Research," in *Proc. of Tridentcom*. IEEE, 2006.

[2] B. White *et al.*, "An Integrated Experimental Environment for Distributed Systems and Networks," in *Proc. of the 5th Symposium on Operating Systems Design and Implementation*. Boston, MA: USENIX Association, Dec. 2002, pp. 255–270.

[3] R. Bajcsy *et al.*, "Cyber Defense Technology Networking and Evaluation," *Commun. ACM*, vol. 47, no. 3, pp. 58–61, 2004.

[4] J. Mirkovic *et al.*, "Automating DDoS Experimentation," in *Proc. of the DETER Community Workshop on Cyber Security Experimentation and Test*. USENIX Association, 2007.

[5] J. Mirkovic *et al.*, "Measuring Denial of Service," in *Proc. of the 2nd ACM Workshop on Quality of Protection*. ACM, 2006, pp. 53–58.

[6] J. Mirkovic *et al.*, "Benchmarks for DDoS Defense Evaluation," in *Proc. of MILCOM*. IEEE, 2006, pp. 1–10.

[7] D. Raychaudhuri *et al.*, "Overview of the orbit radio grid testbed for evaluation of next-generation wireless network protocols," in *IEEE Wireless Communications and Networking Conference*, vol. 3. IEEE, 2005, pp. 1664–1669.

[8] V. Agarwal, "A Scalable Implementation of a Wireless Network Emulator," Master's thesis, University of Utah, 2006.

[9] "Tribe Flood Network 2000." [Online]. Available at http://ca.com/tw/securityadvisor/virusinfo/virus.aspx?ID=8542

[10] "NetStumbler." [Online]. Available at http://www.netstumbler.com/

[11] "Simple Wireless Scanner." [Online]. Available at http://www.swscanner.org/