International Journal of Education and Development using Information and Communication Technology (IJEDICT), 2011, Vol. 7, Issue 2, pp. 46-55.

Development of an Intrusion Detection and Prevention Course Project Using Virtualization Technology

Te-Shun Chou
East Carolina University, USA

ABSTRACT

This paper discusses a project used in a graduate course on intrusion detection and incidents response at East Carolina University. By using virtual machine technology, a virtual network infrastructure was designed for students to simulate the real word attacks in a restricted environment. The project provided students with strong theoretical knowledge and practical experience in the field of intrusion detection and incidents response. The project can be used with both on campus and distance education students.

Keywords: *Intrusion detection and incidents response; virtualization; computer attacks; distance education*

I. INTRODUCTION

Due to the progressive growth of sophisticated computer attacks, it has been a challenge to teach intrusion detection and incident response technology. The students need to understand the behavior of novel attacks' and up-to-date intrusion detection and prevention technologies. The students also need to actually experiment with methods that attackers use to exploit vulnerabilities of computer systems. Our response was to design a project for a graduate intrusion detection and incidents response course from both intruder's and protector's points of view. The experimental environment was built using virtualization technology.

It is possible to build a physical network environment to conduct network intrusion and prevention experiments. The physical network requires a collection of computers, network devices, intrusion detection and prevention systems (IDPS), and components interconnected with each other using hard wiring cables. The hardware typically includes servers, hubs, switches, bridges, routers, and IDPS sensors.

An example of this environment was demonstrated by Ho, Mallesh, and Wright. when they created an ASCENT security teaching lab for both graduate and undergraduate students at the University of Texas at Arlington (Ho, Mallesh, and Wright 2009). Their ASCENT lab consisted of five Dell desktop computers, seventeen Lenovo laptops, three switches, two Cisco routers, and VPN boxes. This approach provides students with an actual network to carry out intrusion detection experiments. However, there are challenges to implementing such an attractive system. One of these concerns cost. The necessary equipment is expensive and resources may not be available. Configuring such a network is time consuming to physically set up all of the network devices. When everything configured, it also requires professionals to take care of the health of network.

The approach taken in this project substituted virtualization technology in place of physical equipment. Instead of using real physical equipment, virtualization technology was employed to build a network with multiple virtual machines. Within a single physical host machine, multiple virtual machines were created and operated simultaneously. In each virtual machine, applications and services were implemented and the virtual machine was able to execute the code just as a

normal physical machine would. A major advantage of this approach is that it eases the load of network administration. Any mistake can be easily fixed and the network can always be kept up and running. When new network topology is needed to be changed to conduct desired cyber attack experiments, it can be easily reconfigured.

Because of these advantages, academic educators have increasingly adapted the concept of virtualization in developing network security courses to enhance student learning. The most recent examples can be found in the works of Du and Gaubatz and Tao, Chin and Lin. Du and Gaubatz created a laboratory environment called Security Education (SEED) using virtualization technology (Du and Gaubatz 2010). The SEED lab environment utilized the open-source Linux Operating system and a number of open-source software tools. It was installed on students' personal computer and the department's general computers. The labs have been used in the department's graduate and undergraduate security courses for many years.

The work of Tao, Cin, and Lin took a different approach (Tao, Chin and Lin 2010). These researchers used VMware in supporting hands-on web security education and developing multiple virtual web security lab modules based on the virtual machines (VMware). They built Ubuntu virtual machines with publicly available tools and installed all necessary web servers, application servers and database servers on them so the students could work on the lab modules.

1.1 Overview of The Intrusion Detection And Protection Project

In our project, a virtual network infrastructure was configured and installed on each student's personal computer. This virtual network environment provided students with realistic experiences in an isolated test environment for conducting network intrusion experiments. An advantage of our approach was that it benefits both on-campus and distance education students by enabling them to perform experiments at any time using their own computers. Most importantly, this approach guarantees that all of the crafted malicious activities are confined inside the network. It provides an isolation environment so no sensitive information can be released to the outside real physical system.

Once the virtual network is properly configured, students are asked to utilize a variety of network security tools to exploit vulnerabilities of virtual machines within the network and employ a packet sniffer to capture traffic passing over the network. For example, the security scanner, Nmap, is used for network exploration and hacking (Nmap). The computer security tool, Metasploit Framework, is used for testing the security vulnerabilities (Metasploit Framework). Also the normal traffic is collected via legal network usage such as browsing websites on the Internet and downloading music from FTP servers. Having finished the collection of both normal and abnormal network activities, students can use a packet analyzer to examine the content of collected traffic and therefore edit the appropriate rules for the implementation of IDPS. Finally, the performance of IDPS is evaluated to inspect whether it can effectively identify network intrusions or not.

The objective of this project is to provide students with a comprehensive study of malicious attacks, intrusion detection, and incident response. The project designs a detailed instructional manual so the students can carry out hands-on essential activities in a step-by-step fashion. Those activities include network configuration, real network attacks generation, collection and analysis, and state-of-the-art IDPS implementation and evaluation. The complete procedure not only provides students with a strong theoretical knowledge in the field of intrusion detection and incident response, but also enhances the students' practical skills for advancement in the current and future network security job market.

This paper is organized as follows. Section 2 presents the methodology used in the project. Section 3 demonstrates the attack categories that we researched. Section 4 discusses the result of the project evaluation. Finally, we will present the conclusions in the last section.

II. METHODOLOGY

This project elaborates a complex process of IDPS development from simulations of real world network breaches to IDPS performance evaluation. It might be possible to ask each student to submit one final report that includes all the required course work at the end of the semester. However, unsatisfactory work might be received due to some student's lack of self discipline and waiting until the last minute to finish their assignments. In order to effectively monitor each student's progress and immediately help students to solve the problems they encounter, this project was divided into seven phases. From phases one to six, each student was asked to submit a phase report that provides a detailed explanation of all of the works they completed. The instructor then comments on the report's strengths and weaknesses. Each phase acts as a learning development for students to raise a level of knowledge to a certain task. With successfully completing six phases, students advance their skills and understanding in the field of intrusion detection and incidents response. Lastly, students integrate all the works done together as a final report, the last report in the cumulative process. The final report represents the student's collective knowledge throughout the entire semester. The six phases and their objectives are shown as follows.

- Creation of an intrusion detection experimental environment
  - to help students recognize the procedure of virtual network installation and configuration
- Attacks recording
  - to help students understand real world network attacks and computer systems' vulnerabilities
- Analysis of attack signatures
  - to help students investigate attack behavior from network traffic
- Generation of intrusion detection rules
  - to help students construct effective intrusion detection rules
- Collection of normal traffic
  - to help students assemble an intrusion detection experimental dataset
- IDPS performance evaluation
  - to help students perform proper evaluation of IDPS
- The final integration
  - To combine everything done in previous phases

2.1 Creating The Intrusion Detection Experimental Environment

The project starts with the creation of a virtual network using the virtualization software VMware workstation 6.0. This allows students to install and configure multiple virtual machines that run different operating systems in one physical machine. For performing intrusion detection experiments, the virtual network generally includes attack host, normal host, victim host, and detection host as shown in Figure 1. With the use of craft programs or existing vulnerability exploitation tools, the attack host is used to launch attacks against the victim host. The normal host is used to generate normal usage traffic. The detection host monitors network segments to find suspect malicious activities. However, in order to simplify the network environment, in this project we only set up two hosts machines (Linux CentOS and Windows XP) within the VMware workstation (Linux CentOS and Windows XP). The Linux machine is used to generate both abnormal and normal traffic as well as act as a detection host. The Windows XP machine is used

to act as the victim host and generate normal traffic.

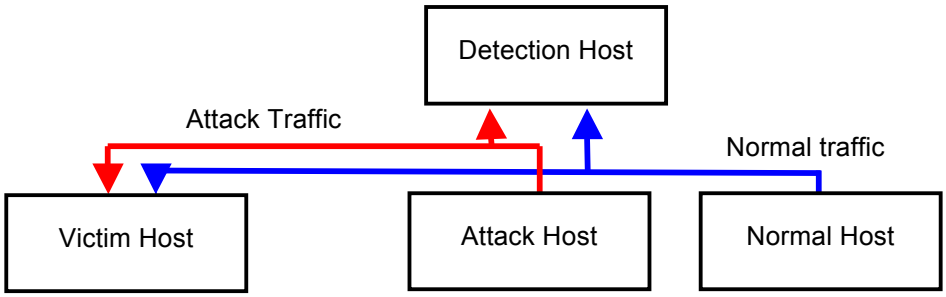

***Figure 1**: The Architecture for intrusion detection experimental environment*

2.2 Attacks Recording

With the rapid growth of Internet based technology, applications of computer networks such as web search and email service are extensively used. In the meantime, networks inevitably become targets of computer attacks. Every day, hackers constantly develop new attack codes to exploit security vulnerabilities of organizations. Once these attacks successfully disable the networks in an organization, the result is that the end users cannot acquire request services as well as the company could lose millions of dollars. While there are many types of attacks, they generally fall into four main classes.

- *Denial of Service* (*DoS*) attacks: Attackers disrupt a host or network service in order to make legitimate users not be able to have an access to a machine;
- *Probe* attacks: Attackers use programs to automatically scan networks for gathering information or finding known vulnerabilities;
- *User to Root* (*U2R*) attacks: Local users get access to root access of a system without authorization and then exploit the machine's vulnerabilities; and
- *Remote to Local* (*R2L*) attacks: Unauthorized attackers gain local access from a remote machine and then exploit the machine's vulnerabilities.

In order to explicate how the attackers take advantage of security vulnerabilities in computer systems or software services, in each attack category one attack is simulated using a particular network security tool. In the *DoS* attack category, Metasploit Framework is used to launch an ultravnc_client buffer overflow attack. In *Probe* attack category, the security scanner Nmap is applied to scan open ports of the target system. In the *U2R* attack category, a backdoor Netcat listener is created in the victim host (Netcat). In the *R2L* attack category, a guessing username/password attack is simulated to test the resistance of the CuteFTP server (CuteFTP). Those attacks are demonstrated in details so that students can follow the steps and accomplish the simulation. This provides students a basic perceptive of how the hackers use diverse approaches to exploit computer security breaches. Afterward, each student is asked to simulate one attack for each category. The student needs to find tools to generate attacks and research their related information on Internet. This further helps students not only better understand cyber attack behavior but also enhance their research and problem solving ability. A few examples of students' self generated attacks are *DoS* attack using UDPFlood, *probe* attack using Angry IP Scanner, *U2R* attack using Backtrack, and *R2L* attack using Hydra (UDPFlood, Angry IP Scanner, Backtrack, and Hydra).

2.3 Analysis of Attack Signatures

Having finished the collection of four groups of attacks, the next step is to analyze the patterns of these attacks and extract signatures from them. In order to acquire enough knowledge to analyze those simulated attacks, the students are first asked to research the collected attacks by studying the related articles on Internet. For example, what is the characteristic of *DoS* attacks? What can buffer overflow cause? How does the ultravnc_client buffer overflow attack computer systems? What is the bid number of ultravnc_client buffer overflow exploit? What is port scanning? How do backdoor attacks work? How many scanning types are there? Is there a three-way handshake establishing a connection between the attack host and target machine's destination port in the beginning of an *U2R* attack?
Next, the students are required to closely inspect those attacks by using the network analyzer Wireshark (Wireshark). By uploading the attack traces into Wireshark, students can investigate the characteristics of packets and extract attack signatures. Those signatures may be present in different potions of packets depending upon the nature of the attack. For example, the signature of a guess password may be found in the payload. A *DoS* attack signature may appear in the IP header. The analysis result will be useful in building rules for Snort IDPS in the following phase.

2.4 Generation of Intrusion Detection Rules

An IDPS is a key element of a network security infrastructure. It examines all network traffic and looks for evidence of suspicious malicious activities. In general, IDPS is classified into two main categories: knowledge-based IDPS and behavior-based IDPS. A Knowledge-based IDPS is typically developed by building a database that models known attack behavior with prior understanding about specific attacks and system vulnerabilities. The system compares network traffic data with those well defined attack patterns, and the possible penetrations to the system can be identified if the data is matched with one of these defined patterns. While knowledge-based IDPS is achieved by modeling known attack signatures, on the contrary, behavior-based IDPS models normal or expected behavior of computer users. It uses soft computing techniques such as fuzzy logic, genetic algorithms, or neural networks to look for malicious activities by comparing the observed network data with acceptable learned behaviors. If the data diverge from the learned normal behavior, the data are classified as attacks.

In this project IDPS Snort is used. Snort is an open-source, network knowledge-based IDPS, which means it includes a set of rules for the signatures of known malicious activities. These rules perform signature matching on network packets in order to discover the threat of potential attacks. Based on the attack signature analysis in the previous phase, students can write proper Snort rules and assess effectiveness of those rules in the IDPS performance evaluation phase.

2.5 Collection of normal traffic

From the decision-based perspective, the goal of intrusion detection is to make decisions on whether network traffic are normal activities or attacks and thus to prevent systems from those hazard attacks. Effective and precise decision making requires collecting a set of network traffic data in advance for analysis. The data consists of a great amount of traffic records with both malicious intrusions and normal computer usages. Based on this set of data, misuse detection specifies well defined attack signatures and anomaly detection constructs acceptable user behavior. Therefore, in this phase the students are required to employ a variety of normal computer activities and store the traffic into a trace file. This is accomplished by downloading and uploading files via the File Transfer Protocol (FTP) server, browsing the Internet through sites such as Google and Yahoo, and monitoring computers' availability using ICMP ping. The collected normal usage traffic is then combined together with the previous generated attack traffic.

This merged single traffic will be used to test the effectiveness of the previously defined Snort rules.

2.6 IDPS Performance Evaluation

By feeding a large amount of network traffic that includes both normal and malicious activities into Snort IDPS, the performance of Snort rules can be evaluated whether they can effectively identify network intrusions or not. In this phase we use Tcpreplay to replay the network traffic trace. Tcpreplay is a suite of tools that can replay previously captured traffic in libpcap format. Here we observe if there are any alerts triggered by defined rules. For evaluating the overall performance of intrusion detection tasks, standard measurements such as *detection rate* (*DR*), *false positive rate* (*FPR*), and *overall classification rate* (*CR*) are used. In general, abnormal activities are expected to be correctly identified and normal activities are anticipated not to be misclassified for an intrusion detection task. Therefore, a higher *DR* and a lower *FPR* are desired. Snort rules are needed to be modified if the performance of Snort IDPS is not satisfied.

2.7 Integration

Finally, all of the results from previous six phases are combined into a single document. This single document is the final technical report for the entire semester.

III. FOUR CATEGORIES OF COMPUTER ATTACKS

3.1. DoS Attacks

For the DoS attack experiment, the Metasploit framework is used to launch an attack from the Linux CentOS host to the Windows XP system. The Metasploit framework is an open source software for use in performing penetration testing, IDPS signature development, and exploit research. Of its 320 exploits and 217 payloads, windows/vnc/ultravnc_client equipped with payload windows/shell_bind_tcp was chosen to exploit ultravnc_client buffer overflow vulnerability of the Windows XP machine.

This is a client buffer overflow attack. The attacker exploits the vulnerability of a system that does not correctly perform a boundary check of the user's input data before copying it to a fixed length memory buffer. Once the vulnerability is found, the attacker can supply excess data into the insufficiently sized memory buffers and therefore possibly corrupt the data and thus make the service crash. Furthermore, the attacker can add executable data into the stream and remotely activate it to gain unauthorized access when the buffer overflows. An example of this attack would be to install a backdoor program on the vulnerable system for future use.

3.2 Probe Attacks

Probe attacks are attacks to explore open vulnerabilities or weaknesses of a network. They aim to gather information on systems within a network in order to lead to access to targeted computers in the future. Among various types of probe attacks, network port scanning is a common way to find out what resources are available on your network. In this experiment, a free security scanner Nmap is used in Linux CentOS host for network exploration of target Windows XP. It divides ports into six states: open, closed, filtered, unfiltered, open|filtered, or closed|filtered. These states give attackers an idea of the status of services in the target computer system.

A variety of scans are provided by Nmap, which includes TCP connect, SYN stealth, FIN, NULL, Xmas Tree, Ping, UDP, IP Protocol, Idle, Ack, Window, RPC, List, Version Detection, Timing and

Hiding Scans. In this experiment, the most common used port scan, TCP SYN scanning, is applied. If the connection to a port is successful, the port is listed as open, otherwise it is said to be closed. The scan result provides the basic port information of a system and the attacker can then look to open ports and vulnerabilities for further exploration.

3.3 U2R Attacks

In a U2R attack, the attacker normally starts with a remote attack to gain access to a vulnerable system. Once the attacker has access at some level as a legitimate user, they will gain a higher level privilege such as administrator or root. This is often done through installing a backdoor program on the compromised system. By using this technique, the attacker can bypass the normal authentication process and easily return to the system for desired activities. Basically backdoors are classified into three basic categories: active, passive and attack-based backdoors (Rudis and Kostenbader 2010). Active backdoors are actively monitored by hackers and can be used anytime whenever they wish to access to the compromised system from the remote systems. Passive backdoors can be triggered by time or events and therefore the attackers have to wait for them to happen. They are similar to active ones that they can establish access into the compromised network for sending data out and receiving acknowledgements and/or commands from the remote systems. Attack-based backdoors could be classified as the "unknown backdoors". They are generally caused from the attackers using the buffer overflow technique to exploit vulnerabilities of poorly-written programs and therefore gain administrator or root level access to the compromised system.

In this experiment a U2R attack is conducted by installing an active backdoor on the target Windows XP system and connecting the attack Linux CentOS host to the victim's http port. Internet Information Services (IIS) is installed in the victim's machine and the default port is 80. After the backdoor is open on port 80 of the target system, the attacker in the remote host can gain the access to the command shell and execute commands such as cd, dir, and mkdir on the victim machine. The entire process is done by creating a Netcat backdoor listener in Windows XP and running Netcat as client mode in Linux CentOS.

3.4 R2L Attacks

For protecting network services, systems in the network always use the authentication technique to prove users' identities by providing their usernames and passwords. In general, people do not create strong passwords so that the attackers have chances to apply the brute force attack or dictionary attack technique to break those bad passwords. The objective of R2L attack experiment is to simulate guessing username/password attack. It starts with running FTP server on the victim Windows XP host, and then the server is connected to the attack Linux CentOS host using a web browser. Once the communication channel is established, the guessing username/password attacks are simulated by entering incorrect information on the client machine. The entire course of attacks is recorded on the victim machine with Wireshark and the packet capture file is saved for future analysis.

IV. PROJECT EVALUATION

An online survey with thirteen individual questionnaires, Table 1, was designed for students' access in the end of fall 2010 semester. The objective of the survey was to evaluate the project's effectiveness in order to improve the project manual for future use. In the design of the survey, we employed eight questions regarding technical issues such as system installation and attack signature analysis. A five-level Likert scale was used. Available responses were: strongly disagree, disagree, neutral, agree, and strongly agree. In order to investigate attitudes of the

respondents toward each question, we coded the responses accordingly: strongly disagree = 1, disagree = 2, neutral = 3, agree = 4, and strongly agree = 5. Table 1 shows the questions in the survey. Table 2 shows the descriptive statistics result. Totally thirteen questionnaires are successfully collected at the end of the course.

***Table 1**: Survey questions*

No.	Question
1	I know how to use both Windows and Linux operating systems.
2	I have no difficulties installing multiple virtual machines on virtualization software.
3	I know how to configure virtual machines in a virtual network.
4	I know how to generate computer attacks to attack vulnerable victims.
5	By inspecting network traffic, I can find possible attack activities with the use of the packet analyzer.
6	After completing the report, I have a better understanding of the signatures of difference attacks.
7	According to the network traffic analysis results, I can write effective intrusion detection rules for IDPS Snort.
8	After completing the report, I have a better understanding of the entire process of IDPS design.

***Table 2**: Survey statistics result*

Question	Strongly Disagree (1)	Disagree (2)	Neutral (3)	Agree (4)	Strongly Agree (5)	N/A	Mean	Standard Deviation
1		1 (7.69%)		5 (38.46%)	6 (46.15%)	1 (7.69%)	4.33	0.89
2	1 (7.69%)	1 (7.69%)	1 (7.69%)		10 (76.92%)		4.31	1.38
3		1 (7.69%)	2 (15.38%)	1 (7.69%)	8 (61.54%)	1 (7.69%)	4.33	1.07
4			2 (15.38%)	7 (53.85%)	4 (30.77%)		4.15	0.69
5			1 (7.69%)	9 (69.23%)	3 (23.08%)		4.15	0.55
6				5 (38.46%)	8 (61.54%)		4.62	0.51
7			3 (23.08%)	9 (69.23%)	1 (7.69%)		3.85	0.55
8		1 (7.69%)	1 (7.69%)	6 (46.15%)	5 (38.46%)		4.15	0.90

With Likert scale data, the most frequent response is the best way to illustrate the analysis result. On the subject of virtualization network environment (Q1 to Q3), over 70% of students expressed that they had no difficulties configuring virtual machines and using them. Two students reported: "*I am very familiar with this, but think it is an essential skill for IT professionals and a great idea to include in this project.*" and "*The configuration packages were provided and was able to follow the*

instructions successfully. I would have to do more research in the VMWare program to determine how to configure additional images."

On the subject of attack generation and analysis (Q4 to Q6), most of the students reported that they knew how to apply security exploitation tools to exploit computer system vulnerabilities. Students also showed that the project helped them understanding the use of packet analyzer to inspect network traffic and extract attack signatures from them. "*After completing this project, I do now have quite a bit better understanding of how to do this task.*", "*I can certainly generate attacks using the tools provided and the tools found. I must say it has peaked my interest and I will maintain the virtual environment for testing of new tools and attacks in the future.*", and "*It was very nice to understand the various types of attacks (DoS, Probe, U2R, & R2L), so then creating and understanding those signatures was very helpful in my learning process.*"

On the subject of creating intrusion detection rules (Q7), 78% of students showed that they can write proper Snort rules according to the signatures extracted from attacks. However there are 23% of students showing a neutral attitude toward this question. This indicates that a tutorial of Snort rules might be necessary in future classes.

On the subject of the understating of IDPS design (Q8), 85% of students agree that they have a better understanding of the entire process of IDPS design after completing the report. "*The process as a whole was very cool. I really enjoyed the technical aspects of this project.*" and "*Using the virtual environment helped to understand the flows of traffic in a small controlled space. We were able to capture and identify specific types of traffic without a lot of other network noise. This is necessary to understand the overall architecture of IDS processes.*"

Overall the average of the eight questions was approximately 4, which shows the students had very positive attitude toward the questions. In addition, we asked students to provide one example where they have added to their knowledge from this project. Some of these responses were: "*I learned quite a bit from configuring and creating rules. I also learned a lot during the attack analysis phase.*", "*I really learned a lot about creating Snort rules as I had not experience in this prior to this class. I also really liked how we were shown how to use Metasploit. Overall, I think I have a much better hands-on mentality of intrusion detection.*", and "*I tried out Metasploit to test my own system's vulnerabilities but I was never able to completely gain access over a machine before this class – seeing is believing. I was able to see this happen first-hand during this class.*"

V. CONCLUSIONS

The virtual network environment provides students a location to perform network security related experiments. The environment should be safe enough to keep any possible hazards away. This project builds a virtual network environment using VMware virtualization software. It allows students quickly and easily to build a network for intrusion detection and prevention experiments. It keeps the student's physical machine safe from artificial attacks since all of them are confined inside the virtual network. In the project, students simulated and analyzed a variety of real world network security breaches. Then each of the students wrote proper rules for simulated attacks and performed an IDPS evaluation. Students were able to understand how serious networks attacks are. The experiments were successful in gradually building a solid foundation in the field of intrusion detection and incidents response and helping students become better prepare for career opportunities in this field.

REFERENCES

Ho, J. W., Mallesh, N., and Wright M. 2009, "The Design and Lessons of the ASCENT Security Teaching Lab," Proceedings of the 13th Colloquium for Information Systems Security Education, pp.124-132, Seattle, WA, June.

Du, W., Jayaraman, K., and Gaubatz, N. B. 2010, "Enhancing Security Education with Hands-On Laboratory Exercises," 5th Annual Symposium on Information Assurance, pp.56-61, Albany, NY, June.

Tao L., Chen, L. C., and Lin C. 2010, "Virtual Open-Source Labs for Web Security Education," Proceedings of the World Congress on Engineering and Computer Science, Vol. I, San Francisco, CA, October.

Vmware: http://www.vmware.com/ (Last browsed in August 2011)

Nmap: http://www.nmap.org (Last browsed in August 2011)

Metasploit Framework: http://www.metasploit.com (Last browsed in August 2011)

Linux CentOS: http://www.centos.org/ (Last browsed in August 2011)

Windows XP: http://www.microsoft.com/windows/windows-xp/default.aspx (Last browsed in August 2011)

Netcat: http://netcat.sourceforge.net (Last browsed in August 2011)

CuteFTP: www.globalscape.com/CuteFTP (Last browsed in August 2011)

UDPFlood: http://www.brothersoft.com/udpflood-79017.html (Last browsed in August 2011)

Angry IP Scanner: http://www.angryip.org/w/Home (Last browsed in August 2011)

Backtrack: http://www.backtrack-linux.org/ (Last browsed in August 2011)

Hydra: http://www.security-database.com/toolswatch/Hydra-password-bruteforcer-updated.html (Last browsed in August 2011)

Wireshark: http://www.wireshark.org (Last browsed in August 2011)

Rudis, B. and Kostenbader, P. 2010, "The Enemy Within: Firewalls and Backdoors," November. http://www.symantec.com/connect/articles/enemy-within-firewalls-and-backdoors (Last browsed in August 2011)