# UNDERSTANDING COMPUTER NETWORK VULNERABILITIES AND SECURITY THREATS VIA PACKET SIGNATURE ANALYSIS

Te-Shun Chou
Department of Technology Systems
East Carolina University

## Abstract

This paper investigates four categories of network attacks used in an intrusion detection and incident response graduate course; they are denial of service (*DoS*) attacks, *probe* attacks, user to root (*U2R*) attacks, and remote to local (*R2L*) attacks. Virtualization technology was applied to simulate the network attacks in a restricted environment. A variety of tools were used to generate, collect and analyze attack traffic traces. One real world attack was simulated in each attack category; they were buffer overflow attack, TCP SYN scanning attack, backdoors attack, and guessing username and password attack. Then, the attacks were analyzed and signatures were extracted for the design of the intrusion detection system (IDS).

## Introduction

Teaching network security is not an easy job. Due to the number of novel and sophisticated attacks that exploit system vulnerabilities within networks that are developed by hackers and crackers every day, it is impossible to teach students every attack that happens in the real world. However, it is possible to teach students the ability of knowing how to analyze hackers' behavior and attack signatures. The best strategy to defend against attacks is to understand your enemy. Hence, in an intrusion detection and incident response graduate course offered by the Department of Technology Systems at the East Carolina University, we designed a project that provides students with hands-on experience in terms of network configuration, real network attacks generation, collection and analysis, and implementation and evaluation of IDS. The complete procedure not only provides a strong theoretical knowledge in the field of intrusion detection and incident response, but also enhances students' practical skills for advancement in the current and future network security job market. In this paper, we focus on the attacks we demonstrated in the designed project.

The goal of this paper is to provide a detailed analysis of those four categories of attacks. The experiments simulated attacks that are conducted by hackers in the real world. To run the experiments, a virtualization technique was used in building a network in a single physical host machine. Multiple virtual machines were created for attack generation and collection. In every virtual machine, a variety of network tools and services were implemented. The virtual machines executed the applications just as a normal physical machine would. All of the experimental attacks were confined inside the virtual network. For each attack category, one attack was demonstrated in detailed steps in the project. Furthermore, each student was asked to research and simulate one additional attack in each category. The collected attack traffic traces were analyzed and their attack signatures were extracted. All of the analysis results were then used to generate detection rules for the use of Snort signature-based IDS [1]. It helps the students in expanding their capabilities in building IDS as well as in evaluating the effectiveness of IDS design.

This paper is organized as follows: First we present the project overview. Then we introduce the four computer attack categories we investigated in the project. Followed by a demonstration of the experimental methodology and a discussion of the experiment results. Next we discuss the online survey statistics result. Finally, we present the conclusions and future work in the last section.

## Project Overview

In addition to the Information and Computer Technology undergraduate program offered by the Department of Technology Systems, the department also offers a Master of Science in Technology Systems (MSTS) program that includes seven concentrations. Among the seven concentrations, three are information technology (IT) related: digital communications technology, computer networking management, and information security. Also, the department offers four graduate level certificates and two of them are information technology related: computer network professional and information assurance. With the growth and development over the last five years, the department decided to separate the IT related concentration and certificate programs from existing graduate programs and established the MS degree program in Network Technology (MSNT). The new program will include four concentrations: digital communications technology, computer networking management, information security, and web technologies. The four concentrations will build on 15 semester hour common core courses and include 15 to 18 semester hours of technical courses depending on thesis or non-thesis track. The course "Network Intrusion, Detection and Incident Response" is one of the technical courses in the concentration of information security.

The course is a three-credit course and taught online during every fall semester. The class enrollment is approximately 15 students per semester and we believe that enrollment will increase in both on-campus and online course sections after the MSNT program launch. Most students who enroll in this course are technology professionals employed in industries and government agencies. Students have diverse technical backgrounds, for example, system support technologist, network security analyst, computer system administrator, IT consultant, and high school and community college instructors who teach IT courses. Some have a wide understanding of the course topics, while some are just beginners in the field. In order to meet students' different learning abilities, we provided entry-to-expert level research papers for reading assignments; we asked students to look at articles and use them in a presentation; we also designed a project that includes both theoretical and hands-on learning activities in the field of intrusion detection and incident response. An instructional project manual was designed to demonstrate how to build an IDS in step-by-step fashion. The project was divided into seven phases, they are:

1. Creation of an intrusion detection experimental environment
   - To help students recognize the procedure of virtual network installation and configuration

2. Attacks recording
   - To help students understand real world network attacks and computer systems' vulnerabilities

3. Analysis of attack signatures
   - To help students investigate attack behavior from network traffic

4. Generation of intrusion detection rules
   - To help students construct effective intrusion detection rules

5. Collection of normal traffic
   - To help students assemble an intrusion detection experimental dataset

6. IDS performance evaluation
   - To help students perform proper evaluation of IDS

7. The final integration
   - To combine everything done in previous phases

During the semester, students are required to submit four reports and each report should provide a detailed explanation of all of the works completed. Report 1 includes all the works of phases 1 and 2. Report 2 includes all the works of phases 3 and 4. Report 3 includes

all the works of phases 5 and 6. Report 4 (the final report) includes the information of phase 7, which includes all the information from phases 1 to 6. The project accomplished two objectives.

a. *To build a virtual network environment using virtualization software*
   It was most common to use actual physical equipment to build a network infrastructure for intrusion detection and response experiments. However, thanks to the advancement of virtualization technology, virtual machines can now be configured to build a virtualized network environment. By using only one single host machine, multiple virtual machines can be used in a network and operated simultaneously. This approach saves cost and time in building a network for intrusion detection and prevention experiments, and at the same time keeps each physical machine safe from experimental attacks since all attacks are confined inside the virtual network.

*b. Comprehensive study of intrusion detection and incident response design*
   This project elaborated on the complex process of IDS development that is now used to identify and describe real world security network breaches and suspicious activities. It helped students develop skills in generating, collecting and analyzing both normal and malicious network traffic. It also helped learners expand their capabilities in building IDS as well as in evaluating the effectiveness of IDS design.

## Four Categories of Computer Attacks

The concept of detecting abnormal behavior of computer users was first introduced by Anderson in 1980 [2]. He published a paper, Computer Security Threat Monitoring and Surveillance, and defined that an attack was a specific formulation or execution of a plan to carry out a threat. He classified a threat as a deliberate unauthorized attempt to access information, manipulate information, or render a system unreliable or unusable. Since then, a variety of taxonomy schemes on grouping attacks into categories have been proposed. For example, in 1987 Denning classified abnormal patterns of system usage into eight categories: attempted break-in, masquerading or successful break-in, penetration by legitimate user, leakage by legitimate user, inference by legitimate user, trojan horse, virus, and denial-of-service [3]. In 1988, Smaha divided intrusions into six main types: attempted break-ins, masquerade attacks, penetration of the security control system, leakage, denial of service, and malicious use [4]. Howard summarized the variations of taxonomy of attacks on the Internet from 1989 to 1995 in one of the chapters in his PhD dissertation [5]. In 1996, Sundaram classified the intrusions into the categories of: attempted break-ins, masquerade, penetration of the security control system, leakage, denial of service, malicious use [6]. Dekker defined network security incident as an activity threat which violated an explicit or implicit security policy and classified incidents into the probe, scan, account compromise, root compromise, packet sniffer, denial of service, exploitation of trust, malicious code, and Internet infrastructure attacks in 1997 [7]. In 1999, Lincoln Laboratory at MIT created the KDD99 data set, which is known as the "DARPA Intrusion Detection Evaluation Data Set" [8]. The data set includes thirty-nine types of attacks that are classified into four main categories: *denial of service* (*DoS*) attacks, *probe* attacks, *user to root* (*U2R*) attacks, and *remote to local* (*R2L*) attacks.

### *DoS Attacks*

In the *DoS* attacks, hackers attempt to disrupt a host or network resource in order to make legitimate users not be able to access the computer service. The victim machines can be web server, domain name system server, mail server, and so on. Known websites, such as Yahoo, eBay, Buy.com, CNN.com, E*TRADE and ZDNet have become victims of *DoS* attacks in the past [9].

*DoS* attacks come in a variety of forms and aim at a variety of services. Generally, they are

categorized into three basic types: consumption of scarce, limited, or non-renewable resources, destruction or alteration of configuration information, and physical destruction or alteration of network components [10]. Among them, flooding is the most common way in which the hackers crumble the victim system with the use of an overwhelming number of packets and, therefore, the services of legitimate users are blocked. For example, smurf attack can cause a target system crash by using the vulnerability of ICMP. The hacker sends a large number of ICMP "echo request" packets to the broadcast address and every packet has a spoofed source address of the intended target system. Any machine in the subnets will respond back to the target by sending ICMP "echo reply" packets. If the number of the packets is more than the target system can handle, the result is that the spoofed system can no longer provide service to the real ICMP requests. Another common way to compromize a system is neptune attacks. It is a SYN flood attack that exists in TCP/IP implementation of a network. The hacker simply rapidly sends out a large number of connection requests but never responds to any replies from the system. While the hacker continues to request new connections faster than the system can handle them, the legitimate connection requests cannot be accommodated. In the mean time, the system may run out of memory and even crash.

### *Probe Attacks*

*Probe* attacks are conducted when hackers use programs to automatically scan a large amount of network IP addresses in order to find vulnerabilities that can be exploited. Once any vulnerability is found, the hackers can gain access to the system and start to gather information without authorization. One of the most common *probe* attacks is called port scanning, which allows hackers to scan all ports on network hosts and discover which ones are available for connections. The popular scanning methods include TCP scanning, UDP scanning, SYN scanning, ACK scanning, FIN scanning, ICMP scanning, protocol scanning, and idle scanning. For example, the portsweep attack discovers exploitable communication channels on remote hosts by systematically requesting connections to multiple TCP ports.

### *U2R Attacks*

*U2R* attacks occur when the hacker pretends to be a legitimate user of the system without authorization and then exploits the system's vulnerabilities to get root access of that system. For example, the hacker may exploit a system's vulnerabilities to gain root privileges and install a backdoor program onto a system for future access. The result may cause the system to crash or make the system execute the hacker's program as if it is part of the system's original programs. Another example is phf attack that exploits a security flaw of CGI script on a web server. Once the vulnerability is identified, the hacker can execute local commands on the attacked remote web server.

### *R2L Attacks*

*R2L* attacks occur when unauthorized hackers, through networks, gain local access as users of local machines. The attacks can be launched from anywhere on the internet. Once the hacker has access to the information systems, they can then exploit the machine's vulnerabilities and cause serious damage such as stealing important data or crashing the information systems. For example, an ftp-write attack is when the hacker takes advantage of the misconfiguration of the ftp service to gain local login to the system. A guess-password attack is when the hacker repeatedly tries possible passwords for gaining access to a user's account. Any service that needs password access possibly becomes an attacked target.

## Experimental Methodology

The project starts with the creation of a virtual network using the virtualization software VMware [11]. It allows us to install and configure multiple virtual machines that run different operating systems in one physical

machine. Within a VMware workstation, two virtual machines, Windows XP [12] and Linux CentOS [13], were preconfigured. The Linux CentOS system was used to launch attacks. The Windows XP system acted as a victim and recorded all the traffic generating from the attack host. In order to generate attacks and collect their traffic for analysis, a variety of network tools were installed and configured in both virtual machines. It included Metasploit framework [14], Wireshark [15], Nmap [16], Netcat [17], Mozilla Firefox [18], Information Internet Services [19], and FTP server [20].

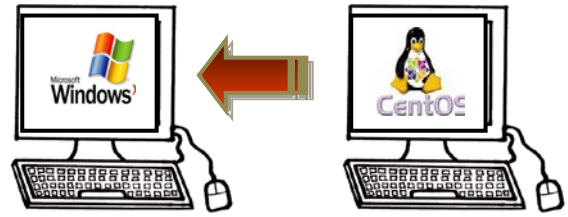


Figure 1. The Experimental Environment.

### *DoS Attacks*

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

This is a client buffer overflow attack. The hacker 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 hacker can supply excess data into the insufficiently sized memory buffers and therefore possibly corrupt the data and thus make the service crash. Furthermore, the hacker can add executable data into the stream and remotely activate it to gain unauthorized access when the buffer overflows. An example is installing a backdoor program on the vulnerable system for future use.

### *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 was used in the 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 hackers an idea of services' status in the target computer system.

A variety of scans are provided by Nmap, which include 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, was 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 hacker can then look to open ports and vulnerabilities for further exploration.

### *U2R Attacks*

In a *U2R* attack, the hacker normally starts with a remote attack to gain access to a vulnerable system. Once the hacker has access at some level as a legitimate user, he/she 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 hacker 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 [21]. Active backdoors are actively monitored by hackers and can be used anytime whenever they wish to access the compromised system from the remote systems. Passive backdoors can be triggered by time or events and therefore the hackers have to wait for them to happen. They are similar to active ones in 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 "unknown backdoors". They are generally caused by the hackers using a buffer overflow technique to exploit vulnerabilities of poorly-written programs and therefore gaining administrator or root level access to the compromised system.

In this experiment a *U2R* attack was 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) was 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 hacker in the remote host can gain access to the command shell and execute commands such as cd, dir, and mkdir on the victim machine. The entire process was done by creating a Netcat backdoor listener in Windows XP and running Netcat as client mode in Linux CentOS.

### *R2L Attacks*

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

## Experimental Results

Figure 2 shows the commands used in Metasploit Framework to start a *DoS* attack on a Windows victim machine. Figure 3 shows part of the packets captured by Wireshark during the attack period of time. After examining the packets, it indicated that the *DoS* attack used TCP port 4444 (krb524 service) as its targeting channel. During the attack, the hacker kept sending SYN packets with random port numbers to the same TCP port of target host. In a one minute period, the hacker sent nearly 120 packets to the same port of target host. Whenever the target machine received a packet from the attack host, it replied with ACK flag as well as RST flag indicating the port is closed. However, the attack host just ignored those responses and kept sending SYN packets out. These packets alternated back and forth and no actual TCP connection was established for further communications. If the service port is open, this SYN flood attack can keep the target busy and therefore makes the target unable to respond to other legitimate requests until the attack ends.

```
[root@localhost ~]# msfconsole

                                  _
                                 | |      o
 _  _  _    _ _|_  __,   ,     _  | |  __    _|_
/ |/ |/ |  |/  |  /  |  / \_|/ \_|/  /  \_|  |
  |  |  |_/|__/|_/\_/|_/ \/  |__/ |__/\__/ |_/|_/
                           /|
                           \|


       =[ metasploit v3.3.3-release [core:3.3 api:1.0]
+ -- --=[ 320 exploits - 99 auxiliary
+ -- --=[ 217 payloads - 20 encoders - 6 nops
       =[ svn r7957 updated 21 days ago (2009.12.23)

Warning: This copy of the Metasploit Framework was last updated 21 days ago.
         We recommend that you update the framework at least every other day.
         For information on updating your copy of Metasploit, please see:
             http://dev.metasploit.com/redmine/projects/framework/wiki/Updating

msf > use windows/vnc/ultravnc_client
msf exploit(ultravnc_client) > set payload windows/shell_bind_tcp
payload => windows/shell_bind_tcp
msf exploit(ultravnc_client) > set rhost 192.168.17.146
rhost => 192.168.17.146
msf exploit(ultravnc_client) > exploit
```

Figure 2. Metasploit Framework.

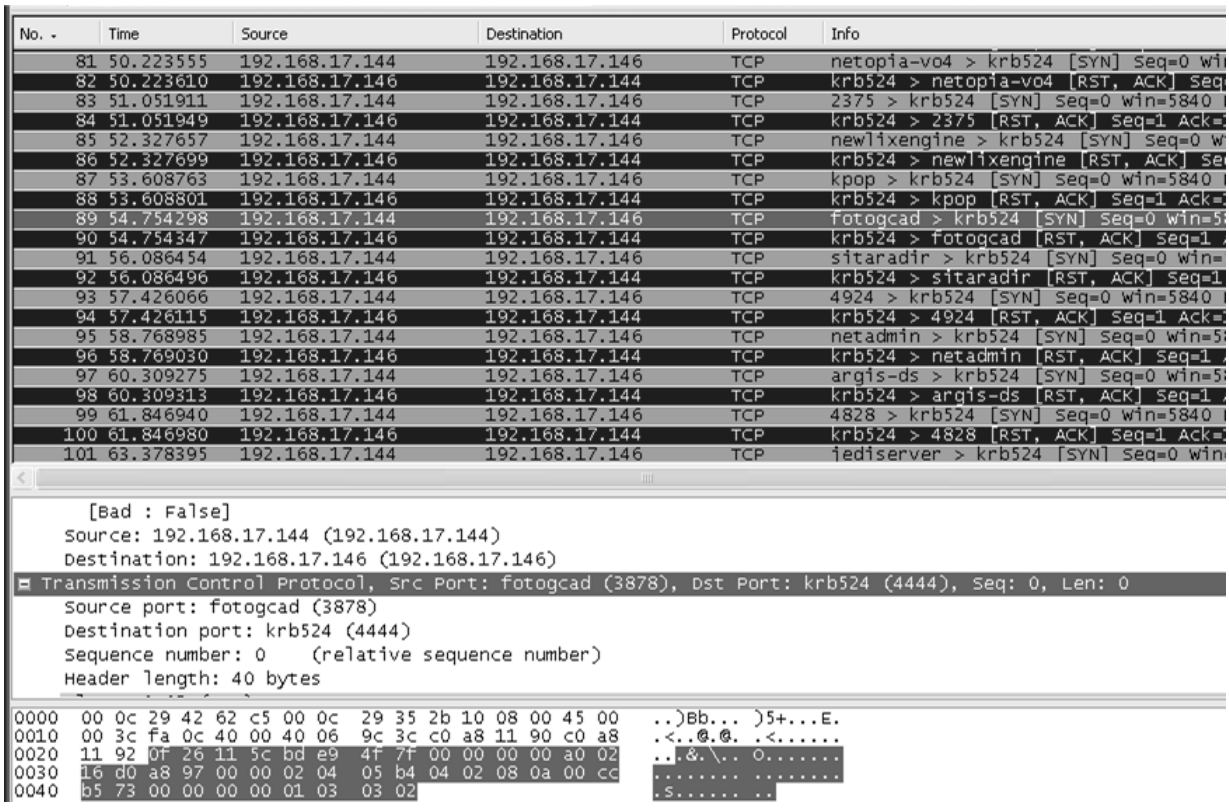


Figure 3. *DoS* Attack.

Next we analyzed a *probe* attack which used a TCP SYN scan to check the port status of the target Windows system. Figure 4 shows the scanning result of open ports on the target machine. The hacker used a half-open scan technique to determine which ports were open and which were closed on its target. SYN packets were sent to the target's port one after another, but a full TCP connection was never established. If a SYN/ACK packet is replied to by the target, it indicates that the port is open and listening. On the other hand the port is closed if a RST/ACK packet is replied. Figure 5 indicates part of the packets during the scanning process that were recorded by Wireshark. It shows most ports of the target system were closed that replied with RST/ACK packets. The two yellow boxes show http and smtp ports were open and SYN/ACK packets were responded. Also, it is noticeable that the hacker used a static source port, 39995, to send all SYN packets to the target system.

```
[root@localhost ~]# nmap -sS 192.168.17.146

Starting Nmap 4.85BETA7 ( http://nmap.org ) at 2010-01-13 15:18 EST
Interesting ports on 192.168.17.146:
Not shown: 990 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
25/tcp   open  smtp
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
443/tcp  open  https
445/tcp  open  microsoft-ds
1025/tcp open  NFS-or-IIS
1027/tcp open  IIS
5000/tcp open  upnp
MAC Address: 00:0C:29:C1:8C:76 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.65 seconds
[root@localhost ~]#
```

Figure 4. TCP SYN scanning.

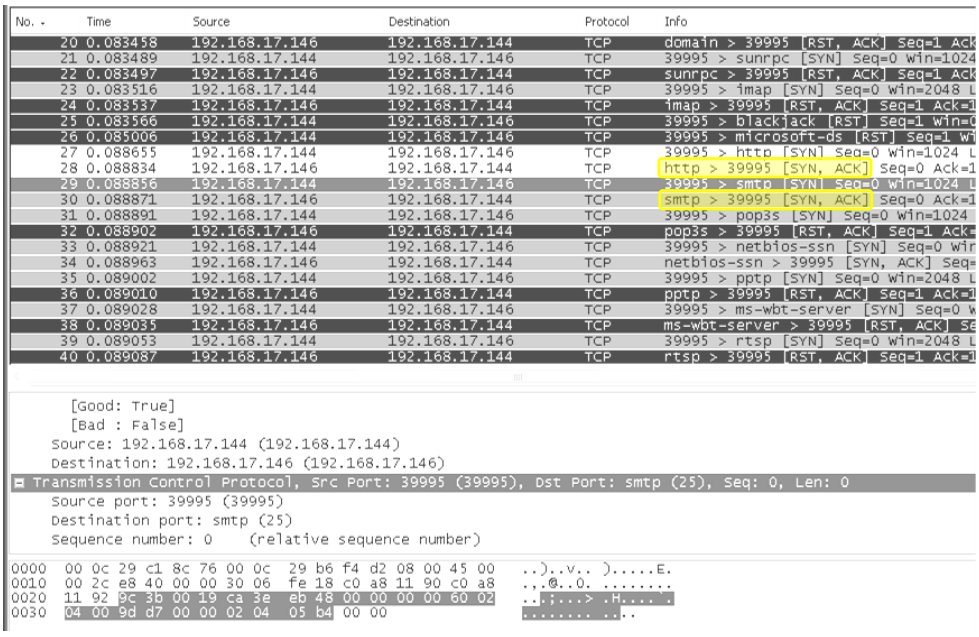


Figure 5. *Probe* Attack.

Figure 6 shows the status and port information of IIS running on the target machine. Figure 7 shows part of the packets of *U2R* attack. In the beginning of the communication, a three-way handshake, a SYN, a SYN/ACK, followed by an ACK, established a connection on an http port between the destination and the source. Then, packet 10 shows that the hacker issued a "dir" command in the remote machine. Packets 11, 13, 15, and 17 show the target sent the queries back to the hacker. The result indicates the hacker successfully bypassed the normal authentication process and obtained access to the target machine undetected.

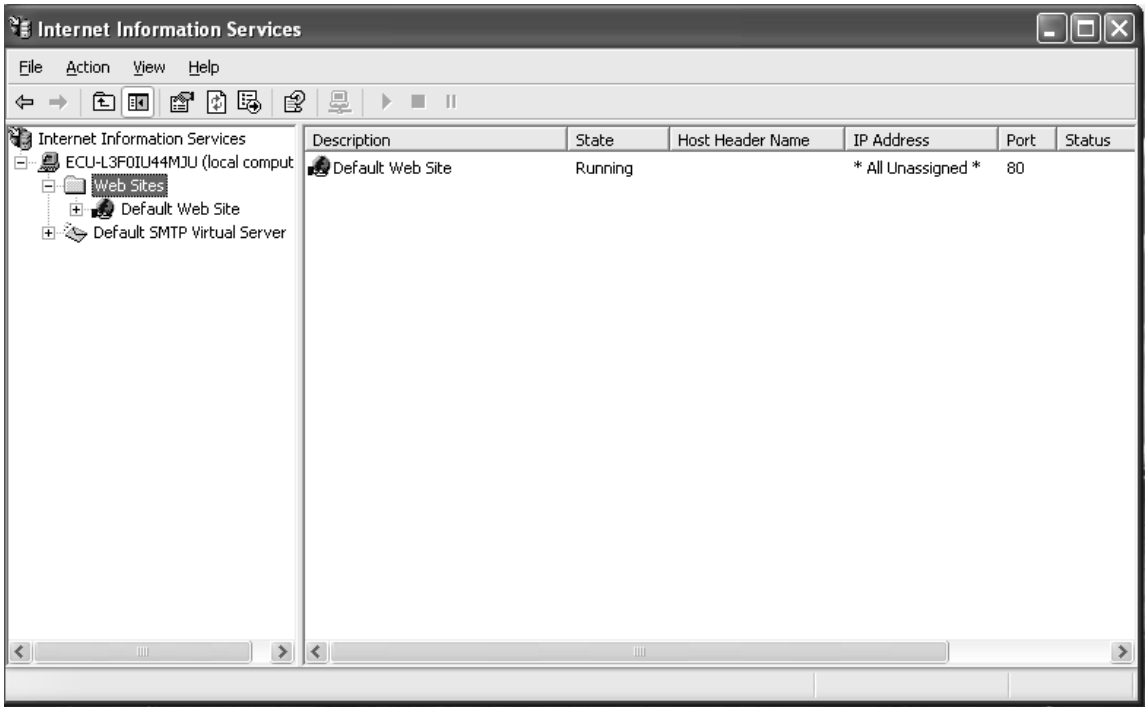


Figure 6. Internet Information Services.

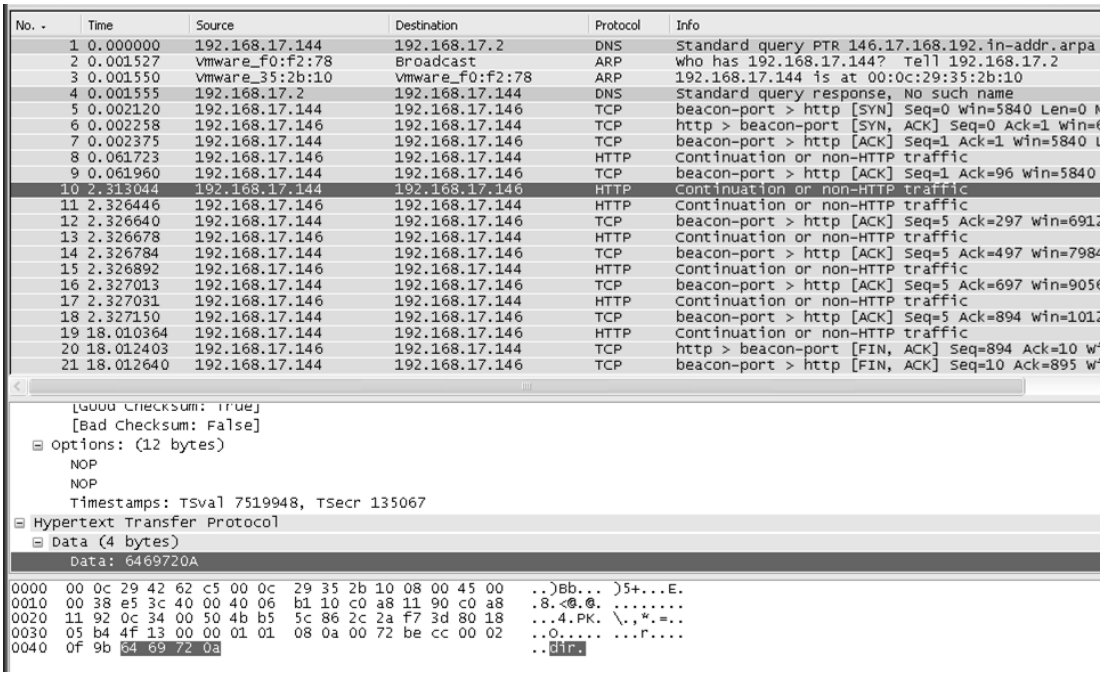


Figure 7. *U2R* Attack.

Finally, a *R2L* attack was simulated by performing a guessing username/password attack in a ftp server that is shown on Figure 8. The attack was analyzed in Figure 9 and the following characteristics in the captured packets were found. First of all, a three-way handshake was communicated between attack host and target machine's port 21. A welcome message was then sent from the ftp server shown on packet 6 followed by a request of the user's login username and password in packet 13. Whenever the user inputs an incorrect username or password, the ftp server stopped its service and the user needed to reconnect to the server again. In the simulation, username "mary" and password "test" were used. During the authentication process between client and server, the result indicates the username and password information is visible with plaintext in the data payload which is shown on the red box of the figure.

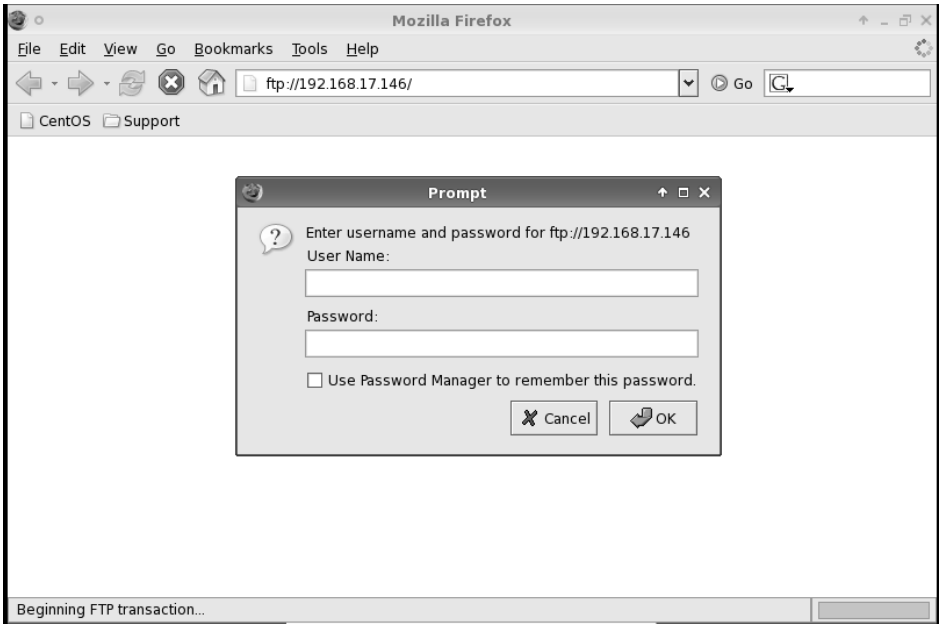


Figure 8. Ftp Login Window.

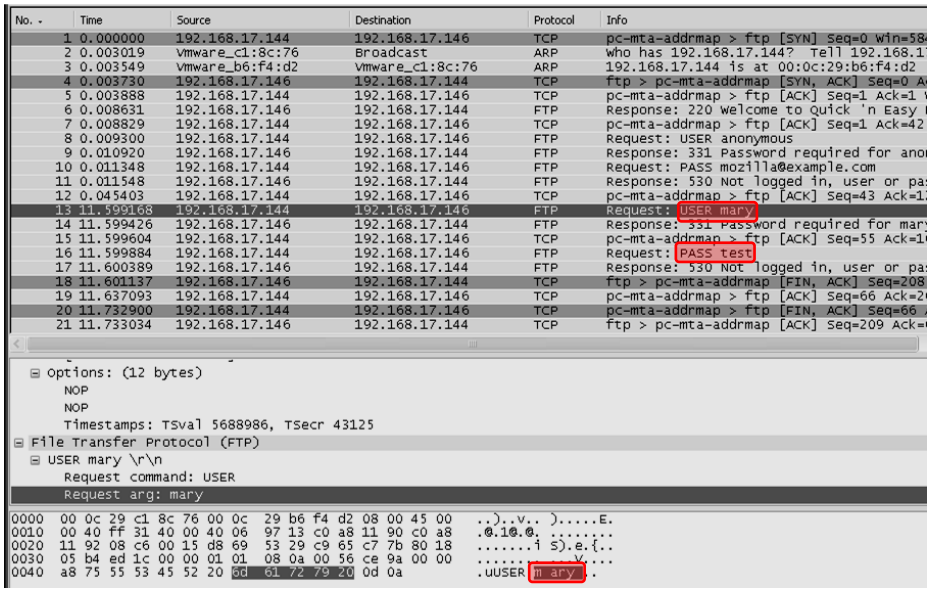


Figure 9. *R2L* Attack.

## Evaluation

At the end of the semester, a survey with eight individual questions was posted online for students' access. The objective of the survey was to evaluate the project's effectiveness in order to improve the project manual for future use. 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.

In the survey, three questions regarding phases 2 and 3 were designed. Table 1 shows the questions and Figures 10 to 12 are the statistical results. A total of twenty-three questionnaires were successfully collected.

Table 1: Three questions regarding attack generation, recording, and analysis

| No. | Questions |
|---|---|
| 1 | I know how to generate computer attacks to attack vulnerable computer systems. |
| 2 | With the help of a packet analyzer, I can find attack signatures by inspecting the network traffic. |
| 3 | After completing the project, I have a better understanding of the signatures of difference attacks. |

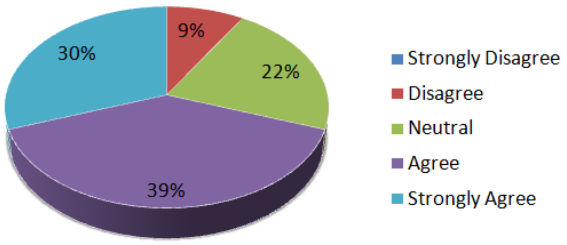


Figure 10. Survey Result of Question 1.

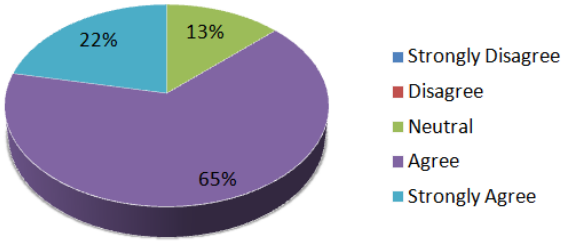


Figure 11. Survey Result of Question 2.

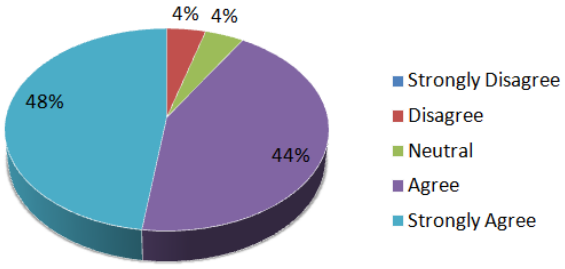


Figure 12. Survey Result of Question 3.

With Likert scale data, the most frequent response is the best way to illustrate the analysis result. Over 90% of students showed they have a better understanding of the signatures of different attacks after completing the project. Students expressed that they knew how to apply security exploitation tools to exploit computer system vulnerabilities and analyzed the attack signatures using a packet analyzer. "*This was my first time to capture and evaluate signatures of different network attacks. As a result of this project, I have a better understanding of the difference of the signatures.*", "*I would say one of my favorite items was learning how to use Metasploit more effectively. For example, I can now launch a brute force password attack and a SYN flood attack.*", "*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.*"

Overall the average of the three questions is 4 points, which shows the students had a 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 responses related to the three questions were: "*This lab exercise greatly improved my ability to define signatures based on a packet capture.*", "*I learned a lot about the different attacks and what each one of them does. I thought the attacks used in this assignment were some of the most common and could find information on.*", "*I learned a great deal about how to use a packet analyzer to better understand network traffic*", "*I learned a lot during the attack analysis phase.*", "*I really liked how we were shown how to use Metasploit. Overall, I think I have a much better hands-on mentality of intrusion detection.*", "*Metasploit, Metasploit, Metasploit. I was intrigued by this program from it's introduction in the course in DoS Attack 1.*", 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.*"

## Conclusions and Future Work

This paper presents research of four categories of network attacks used in a graduate course project in the field of intrusion detection and incident response. In each category, one real world attack was simulated in a virtualization network. The attack traffic was then collected and analyzed in an attempt to extract attack signatures. According to the discovery, a set of rules was designed for the use of the Snort IDS. The project helped students develop skills in generating, collecting and analyzing malicious network activities in real scenarios as well as expand their capabilities in building real IDS and evaluating the effectiveness of IDS design. In the future, more attacks will be included and analyzed, thus enabling students to have a broader understanding of different kinds of network attacks' behavior.

## References


1. Snort network intrusion detection and prevention system, http://www.snort.org/ (accessed January 2013).

2. J. P. Anderson, "Computer Security Threat Monitoring and Surveillance," *Technical Report, James P. Anderson Co.*, Fort Washington, PA, April 1980.

3. E. Denning, "An Intrusion-Detection Model," *IEEE Transactions on Software Engineering*, Volume 13, Number 2, pp. 222-232, 1987.

4. S. E. Smaha, "Haystack: An Intrusion Detection System," in *Proceedings of the Fourth Aerospace Computer Security Applications Conference*, pp. 37-44, Austin, Texas, 1988.

5. J. D. Howard, An Analysis of Security Incidents on the Internet 1989 – 1995, *Dissertation, Carnegie Mellon University,* Pittsburgh, Pennsylvania, 1997.

6. A. Sundaram, "An Introduction to Intrusion Detection," *Crossroads: The ACM Student Magazine*, 2, 4, 1996.

7. M. Dekker, "Security of the Internet," *The Froehlich/Kent Encyclopedia of Telecommunications*, Volume 15, pp. 231-255, New York, 1997.

8. The Defense Advanced Research Projects Agency (DARPA) Intrusion Detection Attacks Database, Massachusetts Institute of Technology Lincoln Laboratory. http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/docs/attackDB.html (accessed January 2013).

9. Piskozub, "Denial of Service and Distributed Denial of Service Attacks," *TCSET*, pp. 303-304, February 2002.

10. Community Emergency Response Teams (CERT) Coordination Center, Denial of Service. http://www.packetstormsecurity.org/distributed/denial_of_service.htm (accessed January 2013).

11. VMware virtualization software. http://www.vmware.com/ (accessed January 2013).

12. Microsoft Windows XP operating system. http://www.microsoft.com/windows/windows-xp/default.aspx (accessed January 2013).

13. Linux CentOS operating system. http://www.centos.org/ (accessed January 2013).

14. Metasploit Framework penetration testing software. http://www. metasploit.com (accessed January 2013).

15. Wireshark network protocol analyzer. http://www.wireshark.org (accessed January 2013).

16. Nmap. http://www.nmap.org (accessed January 2013).

17. Netcat computer networking utility. http://netcat.sourceforge.net (accessed January 2013).

18. Mozilla Firefox web browser. http://www.mozilla.com/firefox (accessed January 2013).

19. Information Internet Services. http://www.iis.net/ (accessed January 2013).

20. Quick 'n Easy FTP Server. http://www.pablosoftwaresolutions.com/html/quick__n_easy_ftp_server.html (accessed January 2013).

21. B. Rudis and P. Kostenbader, "The Enemy Within: Firewalls and Back-doors, " *SecurityFocus*, June 2003. http://www.securityfocus.com/infocus/1701 (accessed January 2013).

## Biographical Information

Te-Shun Chou is an Assistant Professor in the Department of Technology Systems at East Carolina University. He received his Bachelor degree in Electronics Engineering and both Master and Doctoral degrees in Electrical Engineering at Florida International University. He teaches both on campus and online courses. He has been serving as a technical program committee member, publicity chair, session chair, reviewer, and editor at conferences and for journals. He is an editor of the book "Information Assurance and Security Technologies for Risk Assessment and Threat Management: Advances" published by IGI Global. His research interests include network security, intrusion detection and incident response, wireless communications, machine learning, and technology education.