Journal of Information Technology Education: Innovations in Practice **Volume 12, 2013**

Cite as: Trabelsi, Z., & Ibrahim, W. (2013). A hands-on approach for teaching denial of service attacks: A case study. *Journal of Information Technology Education: Innovations in Practice*, *12*, 299-319. Retrieved http://www.jite.org/documents/Vol12/JITEv12IIPp299-319Trabelsi1193.pdf

A Hands-on Approach for Teaching Denial of Service Attacks: A Case Study

Zouheir Trabelsi and Walid Ibrahim
College of Information Technology, UAE University, Al-Ain, UAE

trabelsi@uaeu.ac.ae, **walidibr@uaeu.ac.ae**

Executive Summary

Nowadays, many academic institutions are including ethical hacking in their information security and Computer Science programs. Information security students need to experiment common ethical hacking techniques in order to be able to implement the appropriate security solutions. This will allow them to more efficiently protect the confidentiality, integrity, and availability of computer systems and assets.

This paper presents a case study of the implementation of comprehensive ethical hacking hands-on lab exercises, which are fundamental to security education. The exercises are about three common Denial of Service (DoS) attacks, namely, the Land, the TCP (transmission control protocol) SYN (synchronization) flood, and the Teardrop attacks. DoS attacks are important topics for security courses teaching ethical hacking and intrusion detection techniques. The paper discusses also common defense techniques for detecting DoS attacks, including Intrusion Detection Systems (IDS) and Software tools. Snort tool is used as the IDS defense solution during the hands-on lab exercises. The learning objective of the hands-on lab exercises is for students to learn how to implement and detect the DoS attacks in an isolated network laboratory environment.

Adding ethical hacking to an information security curriculum raises a variety of ethical and legal issues. Some students will use the acquired offensive hands-on skills in inappropriate and sometimes illegal ways. Hence, students may threaten their careers, hurt others, and put their institution's entire information security program at risk. Also, schools and educators may be held liable for the actions of their students. To contribute to improving the chances of having a successful and problem free information security programs that teach ethical hacking techniques, the paper lists a number of steps that should be taken by schools and educators to ensure that students are responsible for their actions and educate students on the consequences of any misconduct.

The impact of offering the exercises on the students' performance in terms of achieving the course outcomes is also discussed. The course assessment results show that the offered hands-on lab exercises allowed students to better anatomize the attacks and assimilate the concepts learned from the lecture. The students have learned better with the exercises which had a positive effect on their performance.

An anonymous questionnaire was administered to students who participated in the hands-on lab exercises to measure their satisfaction level and collect their

Editor: Thomas Connolly

feedback regarding the discussed hands-on lab exercises. The results of the questionnaire showed that more than 85% of the students who answered the questionnaire believed the exercises to be useful and helped them better understand the underlying theoretical concepts associated with DoS attacks.

Keywords: Information security curriculum, DoS attacks, Hands-on lab exercises, Ethical hacking, Schools and educators liability.

Introduction

Defensive techniques with hands-on lab exercises are popular inclusions in security education. However, there is growing interest in offensive techniques which were originally developed by hackers (Bishop, 1997; Brutus, Shubina, & Locasto, 2010; Hill, Carver, Humphries, & Pooch, 2001; Ledin, 2011; Mullins et al., 2002; Trabelsi, 2011; Yuan & Zhong, 2008). In fact, ethical hacking techniques are central for better understanding the ways in which security systems fail. Teaching ethical hacking techniques is becoming a necessary component of computer security curriculum as it yields better security professionals than other curriculums teaching defensive techniques alone (Arce & McGraw, 2004; Arnett & Schmidt, 2005; Dornseif, Gärtner, Holz, & Mink, 2005; Frincke, 2003; Mink & Freiling, 2006; Vigna, 2003). Many academics and industry practitioners feel that the best way to prepare system defenses is to understand the attacks that the systems will face (Arce & McGraw, 2004). Students with ethical hacking skills will understand how attacks are designed and launched and will be better prepared to work as security administrators. This will provide them with better job opportunities than students without attacking skills (Logan & Clarkson, 2005).

Nowadays, there is a noticeable need for computer security textbooks and technical papers that describe the implementation of hands-on lab exercises about ethical hacking techniques. To contribute to satisfy the aforementioned need, this paper proposes comprehensive ethical hacking hands-on lab exercises that are essential to security education. The exercises describe in detail how to practically implement three common DoS attacks. Although, there are many ready-to-use DoS attack tools available in the market, most of them do not have an educational component. In contrast, the exercises described in this paper have an educational purpose since their main objective is to teach students how to build their own DoS attack traffic. The exercises allow students to better anatomize the DoS attacks in an isolated network laboratory environment. The proposed exercises can be offered to students during security courses mainly on ethical hacking techniques and intrusion detection, and are designed to accompany and compliment any existing academic press text. The exercises have been integrated in our intrusion detection and response course (SEC455) offered to our senior information security students. The aim of the lab exercises is to provide the students with the required hands-on experience to improve their comprehension of the different DoS attacks.

The paper is organized as follows. The next section describes the implementation of the hands-on lab exercises. The third section discusses the available common defense solutions for detecting the DoS attacks. The fourth section discusses some ethical concerns emerged when teaching ethical hacking techniques and lists steps that should be taken by schools and educators to improve the chances of having a successful and problem free information security program. In the fifth section, the paper evaluates the impact of the proposed lab exercises on student satisfaction and the achievement of the learning outcomes of an information security senior level course. Finally, the sixth section concludes the paper.

Ethical Hacking Hands-on Lab Exercises

The next three sub-sections describe three common DoS attacks, namely, the Land, the TCP SYN flood, and the Teardrop attacks. For each DoS attack, the corresponding hands-on lab exercise implementation is described. The learning objective of the lab exercises is for students to learn how to implement and detect the DoS attacks in isolated network laboratory environment.

The implementation of the hands-on lab exercises requires heavy involvement of the students. At the beginning of each hands-on lab exercise the instructor briefly summarizes the theoretical concepts related to the DoS attacks which have been already taught in the lecture. Then, the instructor provides the students with the required network architecture setting, the necessary tools to generate and sniff the DoS attack traffic, the network devices (switch and/or routers), and the Intrusion Detection System (IDS) tool to be used to detect the generated DoS attacks.

During the hands-on lab exercises, students work in small groups (three or four students in each group) and are asked to perform mainly the following tasks within one hour:

1. Set up the required network architecture
2. Generate the DoS attacks using packet builder tools
3. Sniff the generated DoS attack traffic using sniffer tools
4. Configure the IDS tool to detect the generated DoS attacks
5. Each group, submit a report including mainly:
    a. Screen shots for the generated DoS attack traffic where they show the contents of the packet's fields corresponding to the DoS attacks
    b. Screen shots for the event logs generated by the IDS tool.

Land Attack

Land attack occurs when an attacker sends spoofed TCP SYN packets (connection initiation) with a target host's IP address and an open port as both source and destination. The target host responds by sending the SYN-ACK packet to itself creating an empty connection that lasts until the idle timeout value is reached. Flooding a system with empty connection requests will overwhelm it and cause it to deny the services that it offers (Figure 1).

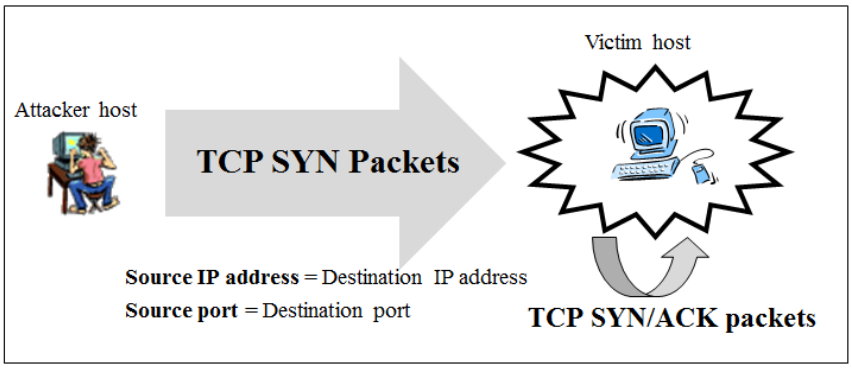

Figure 1: *The Land attack*

Hands-on lab exercise on Land attack

The learning objective of this hands-on lab exercise is for students to learn how to perform Land attack using an IP packet builder tool. A simple LAN network is used in this exercise. That is, three hosts are connected to a switch and each host is assigned a static IP address, as shown in Figure 2. We assume that host "A" is the attack host, and host "B" is the victim host. Host "C" is connected on a monitoring port of the switch (known as SPAN port) to monitor the network traffic exchanged between hosts "A" and "B" using CommView sniffer (CommView Network Monitor, 2013), as a packet analyzer tool. Also, host "C" uses Snort (Snort, 2011), as an IDS software tool, to detect attack traffic.

The lab exercise consists of the following three steps:

- *Step 1*: Configure a SPAN port on the switch
- *Step 2*: Generate Land attack traffic using an IP packet builder tool
- *Step 3*: Sniff Land attack traffic

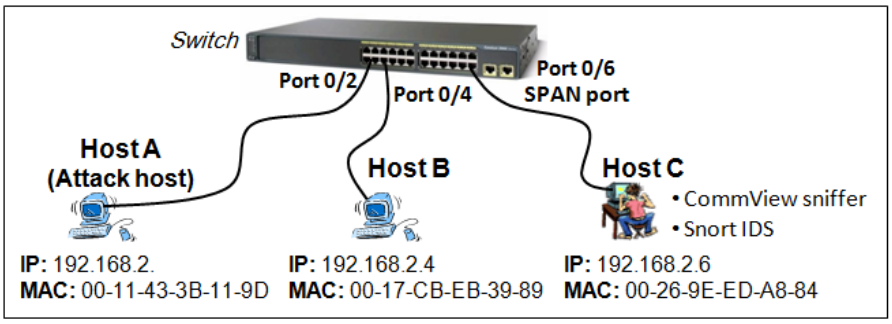

Figure 2: *Network architecture*

Step 1: Configure a SPAN port on the switch

Since each switch manufacturer defines its specific set of steps and commands to configure a SPAN port, in this paper Cisco Catalyst 4500 Series Switch (Cisco, 2013) is used as an example. In Windows XP environment, to configure a SPAN port on a Cisco switch, simply perform the following steps:

- Connect a host to the console port on the switch.
- Start a Terminal Application program (For example: HyperTerminal) in the host.
- Run the following commands to configure a SPAN port and save the configuration:

  Switch> enable //enter the enable command to access privileged EXEC mode
  Switch# configure terminal
  Switch(config)# monitor session 1 source interface fastethernet 0/2 both
  Switch(config)# monitor session 1 source interface fastethernet 0/4 both
  Switch(config)# monitor session 1 destination interface fastethernet 0/6
  Switch(config)#exit
  Switch#copy running-config startup-config

The above commands configure the Cisco switch to:

1. Monitor all the traffic from the following two sources:
- *Host "A" [Fastethernet 0/2] - [Both]: Incoming & Outgoing Traffic*
- *Host "B" [Fastethernet 0/4] - [Both]: Incoming & Outgoing Traffic*

2. Deliver a copy of them to one destination: *Host "C" [Fastethernet 0/6]*

Step 2: Generate Land attack traffic using an IP packet builder tool

Land attack traffic is built from spoofed TCP SYN packets with the target host's IP address and the open TCP port as both source and destination. Figure 3 shows example values of the main fields of a Land attack packet.

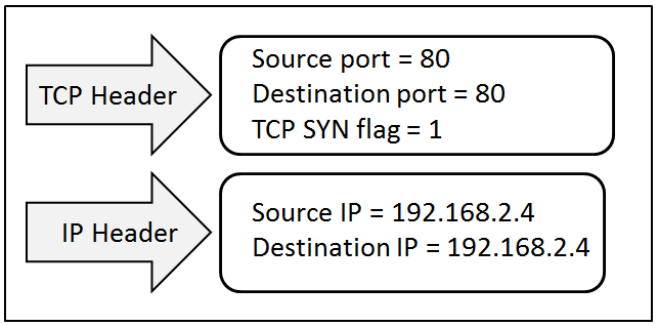

Figure 3: *An example of a Land attack packet*

A packet generator tool can be used to customarily build the Land attack traffic packets. This can be done by using an online command tool, such as *FrameIP Packet Generator* (FrameIP Packet Generator, 2013), or a more friendly and easy to use GUI tool, such as *Engage Packet Builder* (Engage Packet Builder, 2013) or *CommView Visual Packet builder* (CommView Network Monitor, 2013). For instance, from the attack host "A" and using CommView Visual Packet Builder, Figure 4 shows a spoofed TCP SYN packet used to generate Land attack traffic. The packet has the source IP address equals to the destination IP address (Host B's IP: 192.168.2.4), and the source port equals to the destination port 80. The destination MAC address is set to the MAC address of the target Host "B".

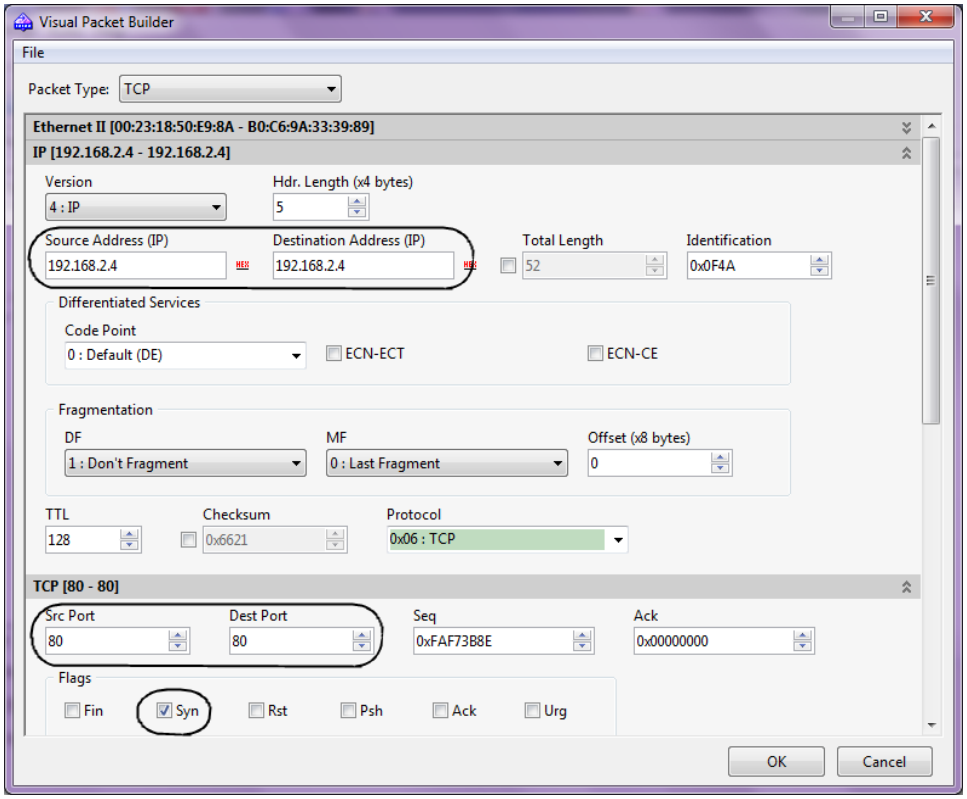

Figure 4: *Spoofed TCP SYN packet for generating Land attack traffic*

Step 3: Sniff Land attack traffic

The aim of this step is to verify that the intended traffic has been generated adequately. Host "C" uses CommView sniffer to capture the exchanged traffic between hosts A and B. Figure 5 is a screenshot showing the captured Land attack traffic generated in Step 1. It shows clearly that the victim host B (192.168.2.4) has been flooded with Land attack packets.

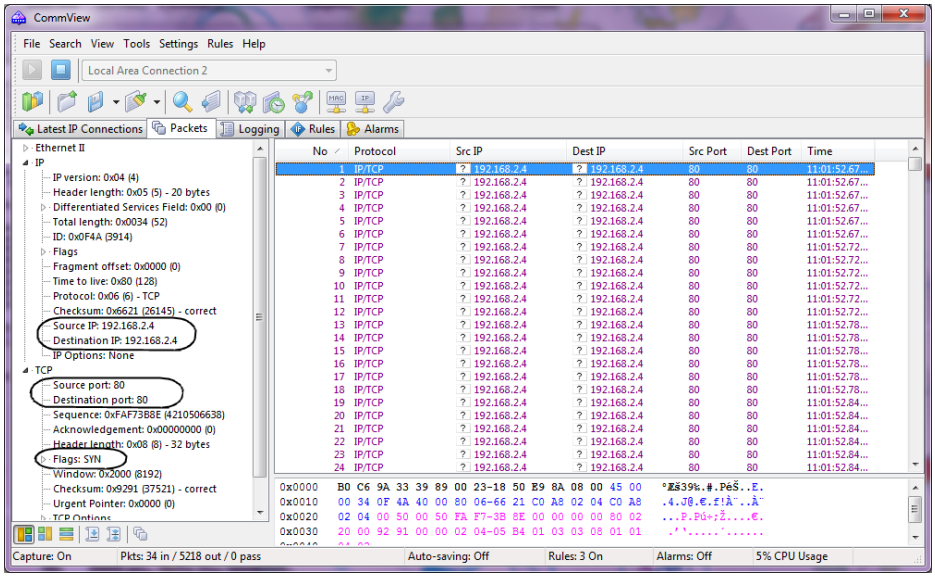

Figure 5: *Land attack traffic captured by CommView sniffer*

TCP SYN Flood Attack

A TCP SYN flood attack occurs when a host becomes so overwhelmed by TCP SYN packets initiating incomplete connection requests such that the host can no longer process any new legitimate connection requests. When a client system attempts to establish a TCP connection to a server, the client and server exchange a sequence of messages and the process is known as the three-way handshake. The client system begins by sending a SYN message to the server. The server acknowledges the SYN message by sending a SYN-ACK message to the client. The client then finishes the establishing of the connection by responding with an ACK message. The connection between the client and the server is then opened, and the service-specific data can be exchanged between the client and the server.

The potential of abuse arises at the point when the server system sends the SYN-ACK message back to the client and before it receives the final ACK message. This situation is referred to as a half-opened connection. The server usually keeps in its memory a data structure that describes all the pending connections. Since this data structure has a finite size, it can be easily overflowed by intentionally creating too many partially-opened connections (Figure 6). Creating half-opened connection is easily accomplished with IP spoofing. The attacker's host sends SYN messages to the victim's server. The messages appear to be legitimate to the server; however, the source address is spoofed to a host that is not connected to the network. This means that the victim server will never receive the final ACK message. Since the source address is spoofed, there is no way to determine the identity of the true attacker when the packet arrives at the victim system.

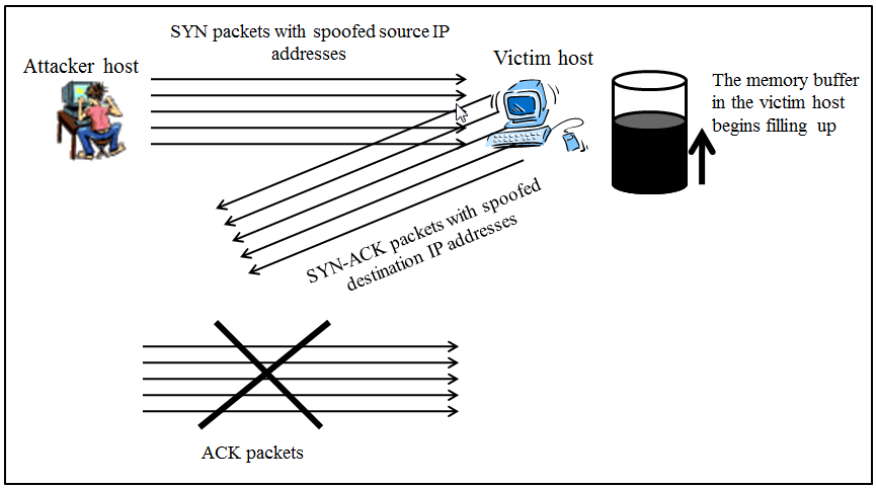

Figure 6: *The TCP SYN flood attack*

Hands-on lab exercise on TCP SYN flood attack

The learning objective of this hands-on lab exercise is for students to learn how to perform TCP SYN flood attack using an IP packet builder tool. The exercise uses the same network architecture described in the previous exercise, and consists of the following steps:

- Step 1: Generate TCP SYN flood attack traffic.
- Step 2: Sniff TCP SYN flood attack traffic.

Step 1: Generate TCP SYN flood attack traffic

Since Host "B" (192.168.2.4) is the victim host, the TCP and IP headers of TCP SYN flood attack packets should be set to the values shown in Figure 7. That is, the source IP address should be set to a spoofed or random IP address, and the destination port should be set to a number of an open TCP port in the victim host.

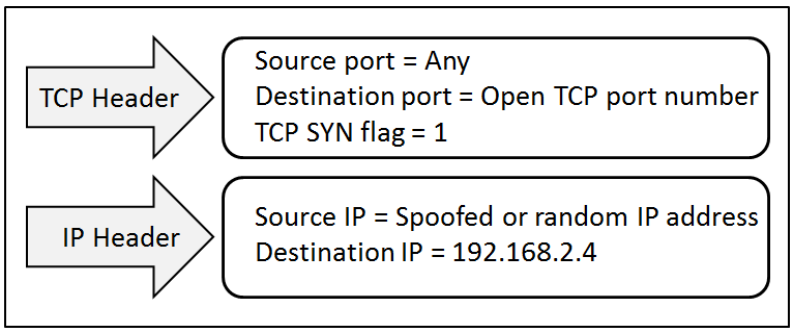

Figure 7: *An example of a TCP SYN flood attack packet*

The attacker at host "A" can use any port scanner tool to identify the list of open TCP ports at the victim host. Then, the attacker can select one open TCP port number and use it as the destination port number in the TCP SYN flood attack packets. For example, Figure 8 is a screenshot showing the result of a TCP port scanning of the target host "B", using Advanced Port Scanner tool (Radmin Advanced Port Scanner, 2013): There are 8 open TCP ports on host "B".

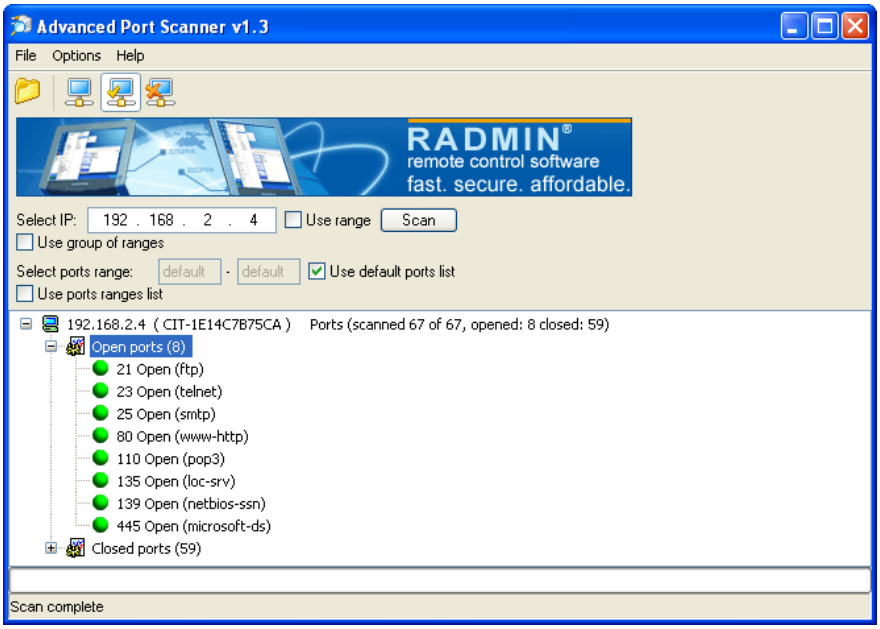

Figure 8: *Port scanning result using Advanced Port Scanner tool*

To generate TCP SYN flood attack packets, the attacker has to use an IP packet builder tool that allows the insertion of random IP addresses in the source IP field and the generation of high packet rates. However, only few packet builder tools support this feature. For example, FrameIP Packet Generator tool has the ability to generate packets with random IP addresses and/or ports numbers and offers high packet rate. Figure 9 is a screenshot showing the results of executing FrameIP tool. That is, the target host with IP address 192.168.2.4 is flooded with TCP SYN packets. Each generated TCP SYN packet has a random fake source port number and a random fake source IP address.

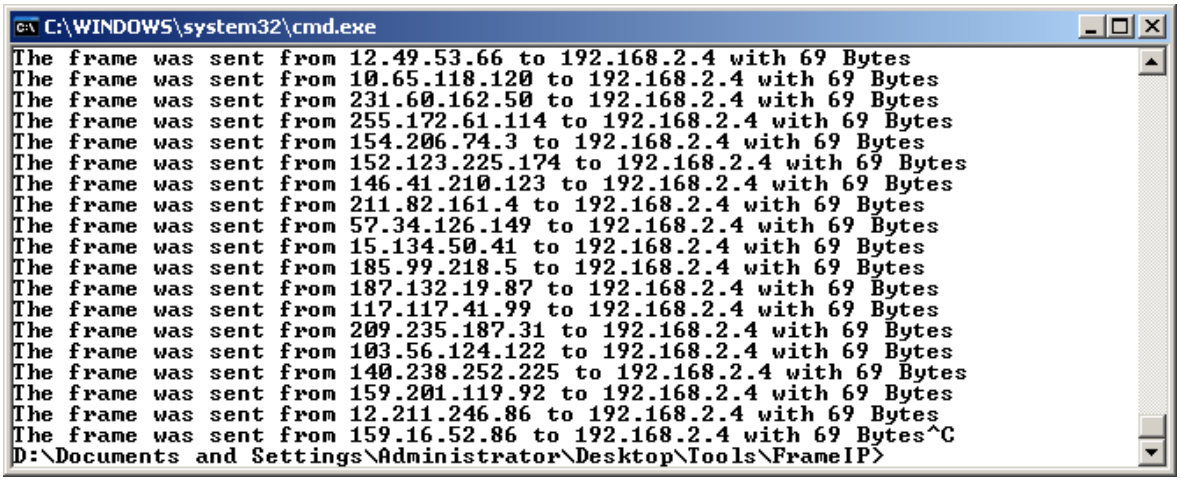

Figure 9: *TCP SYN flood traffic generated by FrameIP tool*

Step 2: Sniff TCP SYN flood attack traffic

At Host “C”, a sniffer can be used to capture the generated traffic. The aim of this step is to analyze and verify that the intended traffic has been generated adequately. For example, using

CommView Sniffer, Figure 10 shows that the victim Host "B" (192.168.2.4) is under TCP SYN flood attack, and the target TCP port is 80.

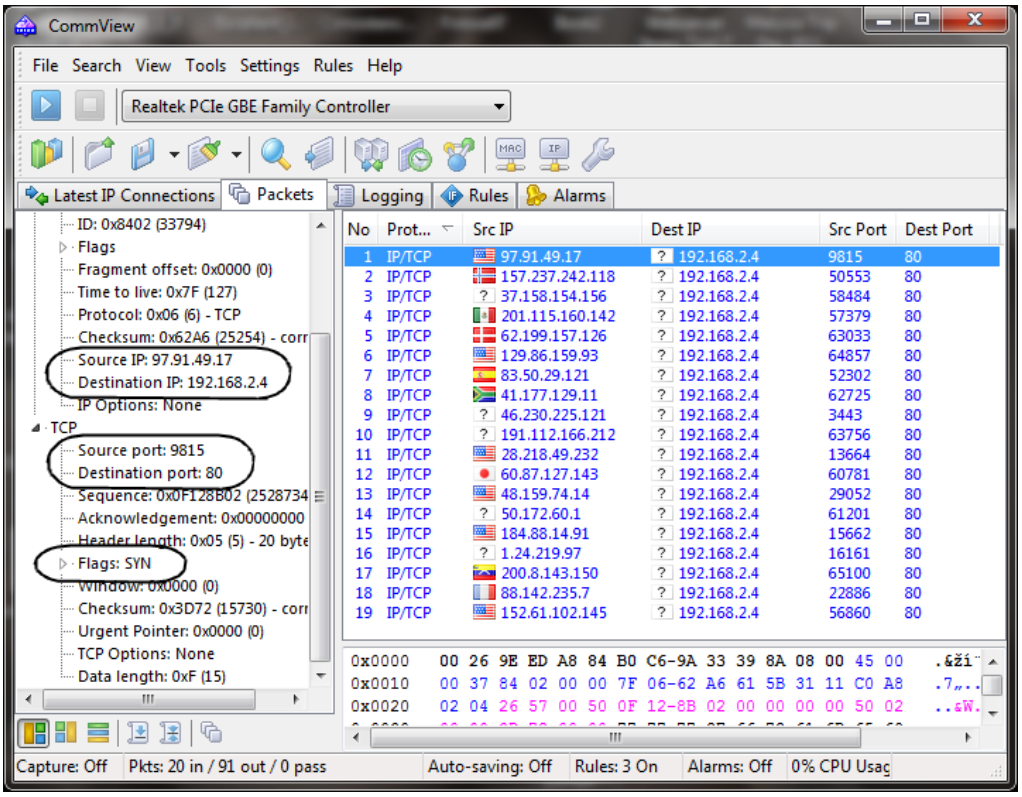

Figure 10: *TCP SYN flood attack traffic captured by CommView sniffer*

Furthermore, at the victim Host "B", the online DOS command "netstat" can be also used to detect TCP SYN flood attack. The command displays the connections that are currently in the half-open state. The half-open state is described as SYN_RECEIVED in Windows and as SYN_RECV in Unix systems.

Teardrop Attack

Teardrop attack targets vulnerability in the way fragmented IP packets are reassembled. Fragmentation is necessary when IP datagrams are larger than the maximum unit of transmission (MUT) of a network segment across which the datagrams must traverse. In order to successfully reassemble packets at the receiving end, the IP header for each fragment should include an offset to identify the fragment's position in the original un-fragmented packet. In a Teardrop attack, packet fragments are deliberately configured with overlapping offset fields causing the host to hang or crash when it tries to reassemble them. Figure 12 shows that the second fragment packet (Packet #2) purports to begin 20 bytes earlier (at 800) than the first fragment packet (Packet #1) ends (at 820). The offset of Packet #2 is not aligned with the packet length of Packet #1. This discrepancy can cause some systems to crash during the reassembly attempt.

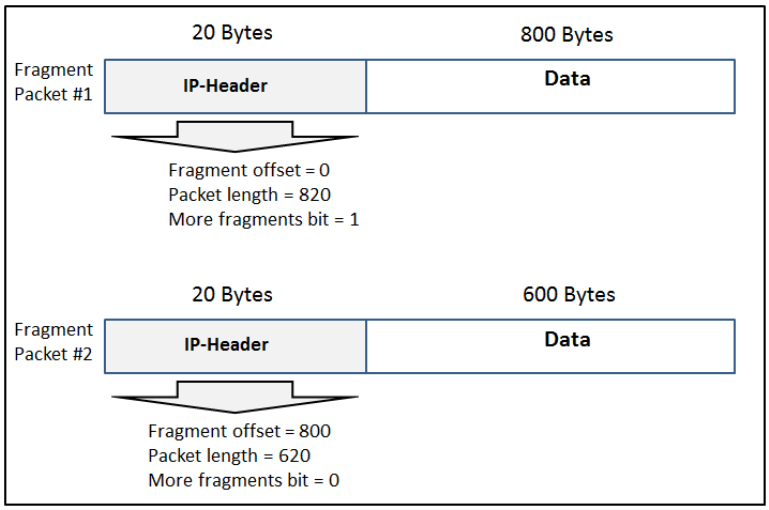

Figure 12: *The Teardrop attack*

Hands-on lab exercise on Teardrop attack

The learning objective of this hands-on lab exercise is for students to learn how to perform Teardrop attack using an IP packet builder tool.

The exercise uses the same network architecture described in the first exercise, and consists of the following steps:

- Step 1: Generate Teardrop attack traffic.
- Step 2: Sniff Teardrop attack traffic

Step 1: Generate Teardrop attack traffic

To generate a Teardrop attack, two fragmented packets are built. The packets have the same IP's ID, which means that they belong to the same original un-fragmented packet. However, offset values are overlapped. Figure 13 illustrate an example of the IP header values for two Teardrop attack packets. To generate these packets, the attacker has to use an IP packet builder tool that allows sending simultaneously more than one packet. However, few packet builder tools, such as the FrameIP Packet Generator, offer such capability.

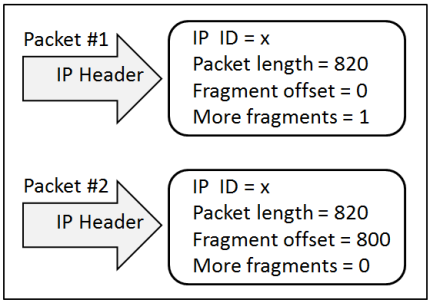

Figure 13: *An example of Teardrop attack packets*

Step 2: Sniff Teardrop attack traffic

The aim of this step is to verify that the intended attack traffic has been generated adequately. The monitoring host "C" uses CommView sniffer to capture the exchanged traffic between hosts "A" and "B". Figure 14 and 15 are screenshots of the captured two Teardrop attack packets (Packet#1

and Packet#2) sent to the victim Host "B" (192.168.2.4). The two fragmented packets belong to the same original un-fragmented packet (IP's ID = 200) and have overlapping offset values, 0 and 20 respectively.

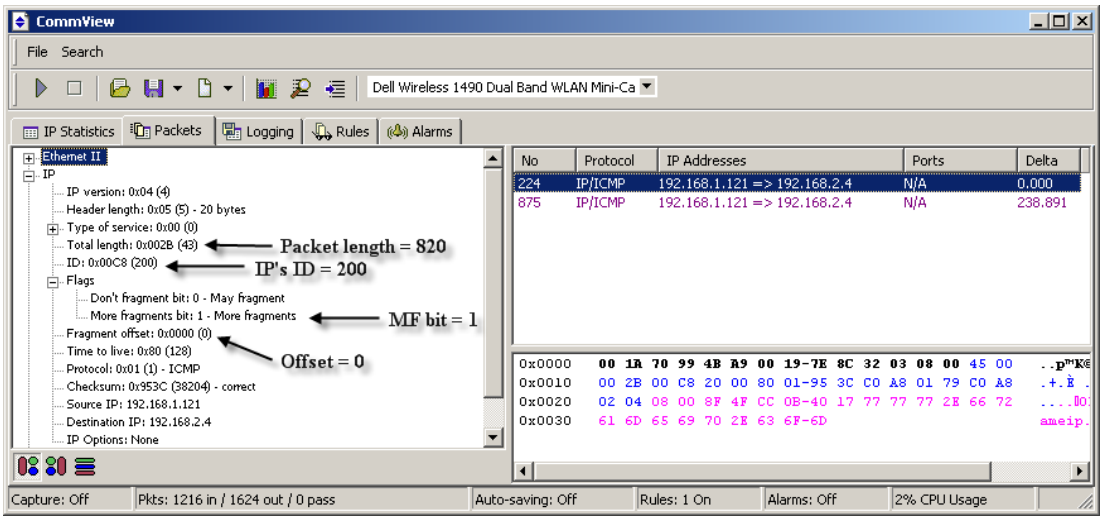

Figure 14: *The first packet of a Teardrop attack (Packet #1)*

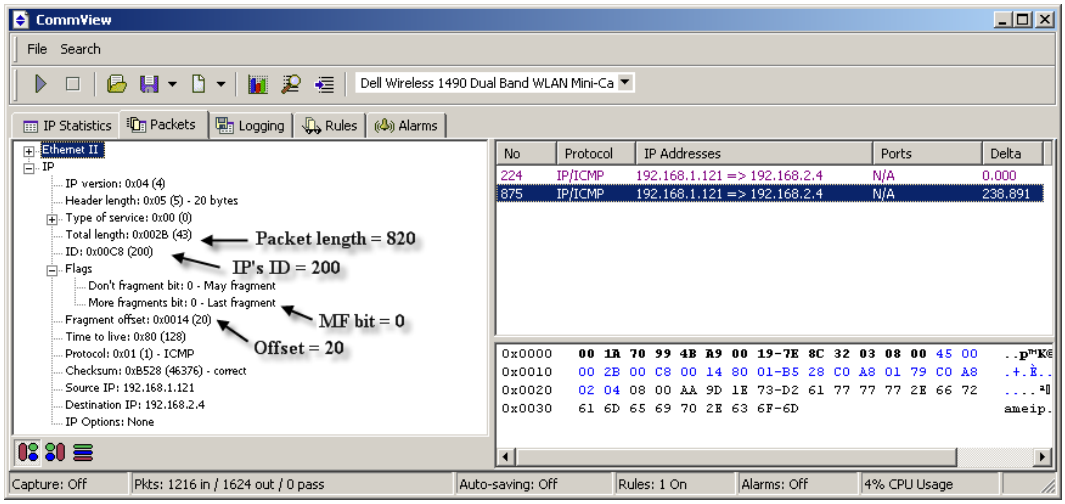

Figure 15: *The second packet of a Teardrop attack (Packet #2)*

Defense Solutions

Defense solutions against DoS attacks are of a great interest because these types of attacks are highly intentional and usually have to be initiated, maintained, and controlled by humans. However, DoS attacks cannot be totally prevented. There is always the chance that the attacker may send to a victim computer system too much data that it could not handle. However, the threat of DoS attacks can be minimized by increasing the network bandwidth and by using vendor patches, firewalls, Intrusion Detection/Prevention systems (IDS/IPS) software tools or hardware appliances, and proper network configuration. Operating systems offer also methods for hardening the TCP/IP protocol stack, which reduces the servers' vulnerability to many common DoS attacks. A modification of the default TCP/IP stack settings is recommended during the process of securing the operating system. For example, steps to harden the TCP/IP protocol stack to make servers

more resistant to the TCP SYN flooding attack are detailed in Burdach (2010) and Microsoft (2011). However, an attacker can always use additional resources to flood a target system or network and/or invent new and unknown types of DoS attacks. The following two sub-sections describe briefly the common IDS/IPS based defense solutions.

IDS/IPS Hardware Devices

IDS/IPS hardware devices are designed to detect and prevent malicious traffic and activities in computer networks. They can be easily configured to detect and prevent common DoS attacks. For example, Figure 16 is a screenshot showing that the Land attack protection and Teardrop attack protection options are enabled at Juniper Networks SSG20 device (Juniper Networks, 2013).

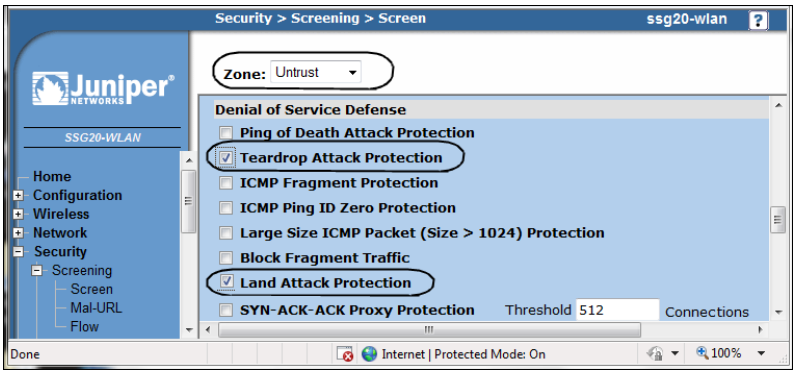

Figure 16: *Land and Teardrop attacks protection options in Juniper Networks SSG20 device*

Figure 17 is a screenshot showing the event log contents in Juniper Networks SSG20 device after detecting Teardrop attack traffic. Event log shows that Teardrop attack traffic has been generated from a host with IP address 192.168.1.133, targeting a host with IP address 192.168.2.4.

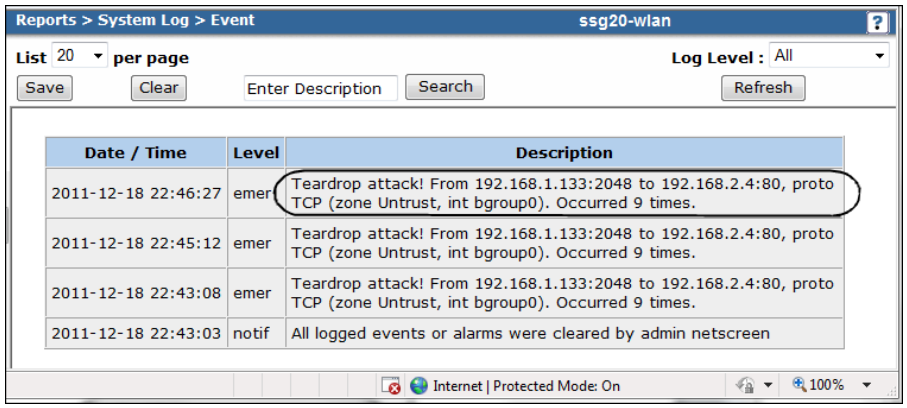

Figure 17: *Event log contents in Juniper Networks SSG20 device after detecting Teardrop attack traffic*

IDS Software Tools

Snort, an open source network intrusion detection system (NIDS), is a good example of an IDS software tool. It can perform protocol analysis and content searching/matching. It can also be used to detect a variety of attacks and probes such as DoS attacks, buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, and much more. The host running Snort should be installed on the SPAN port of a switch to be able to monitor and analyze the network traffic exchanged.

Snort uses a database of rules to detect attacks and malicious traffic. Snort rules are the conditions specified by the network administrator that differentiate between normal Internet activities and malicious activities. Snort rules are made up of two basic parts:

- Rule header: This is the part where the rule's actions are identified. Alert, Log, Pass, Activate, Dynamic, etc. are some important actions used in Snort rules.
- Rule options: This is the part where the rule's alert messages are identified.

For example, the following Snort rule is used to detect TCP SYN flood attack traffic:

alert tcp any any -> any any (msg:"TCP SYN flood attack detected"; flow: stateless; flags:S,12; threshold: type threshold, track by_src, count 3, second 1; classtype: attempted-recon; sid:10002;rev1;).

The following screen shot shows an example of Snort's event log after detecting TCP SYN flood traffic:

```
[**] [1:5000001:2] TCP SYN flood attack detected [**]
[Priority: 0]
01/02-10:50:53.152327 0:1E:B:2C:86:36 -> FF:FF:FF:FF:FF:FF type:0x800 len:0x45
77.238.187.181:60783 -> 192.168.1.3:80 TCP TTL:128 TOS:0x0 ID:44551 IpLen:20 DgmLen:55
******S* Seq: 0x2CDCB70E  Ack: 0x0  Win: 0x0  TcpLen: 20
```

Snort IDS based hands-on lab exercises

Snort is widely used in academia as a tool for teaching network security concepts (Sharma & Sefchek, 2007; Xu, Zhang, Gadipalli, Yaun, & Yu, 2011). During the hands-on lab exercises, the student groups use Snort as a defense solution. For each hands-on lab exercise, student groups are asked to write the appropriate Snort rules to detect the corresponding DoS attack. Each group submits then a report including mainly:

1. Snort rules written to detect the DoS attacks
2. Screen shots for Snort event logs generated after detecting the DoS attacks.

Ethical Concern

The hands-on lab exercises have been offered in our intrusion detection and response course (SECB 455) during the last three years. A major ethical concern has been identified while analyzing the log files of the IDS sensors installed in the university network segments.

For a period of three years, during each semester we used the IDS sensors' log files to collect the number of detected DoS attacks per day targeting the university servers. The collection is mainly focused on the web, emails, FTP, and DNS servers. The collected data shows clearly a significant increase (300% to 750%) in the average number of DoS attacks detected by the university's IDS sensors during the few days following the hands-on lab exercises practice (Figure 18). This is due to the fact that students always attempt to experiment the learned DoS attacks outside the isolated network laboratory environment.

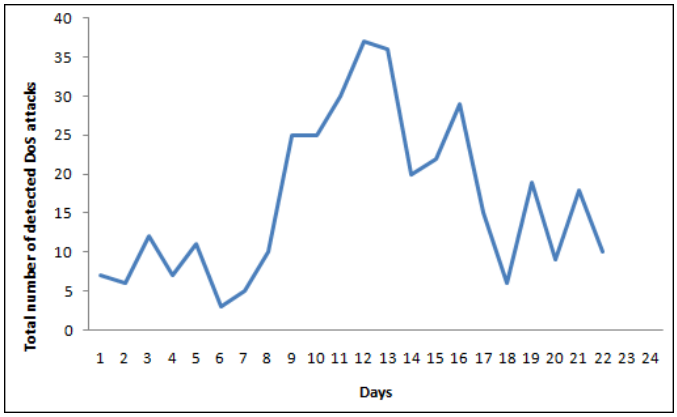

Figure 18: *Evolution of the number of DoS attacks detected by the University's IDS sensors*

At the same time, we conducted a survey to probe the students' behavior after executing the attack lab exercises. The survey results show that most of the students (85%) acknowledged that they had tried the learned DoS attacks outside the isolated network laboratory environment, targeting mainly the university servers. Table 1 shows the result of the survey conducted over the last three years period on about 110 students enrolled in the SECB455 course.

Table 1: Survey results

After the hands-on lab exercises practices, did you experiment the DoS attacks outside the isolated network laboratory environment?	**85% of the students said "Yes"** **12% of the students said "No"** **3% abstained**
If yes, what were your main target systems?	University Web servers (56%) University Email servers (14%) Other university servers (9%) Outside systems (21%)
What were the objectives of the attacks?	For fun (89%) Attempting to slow down the target systems (11%)

This is a dilemma when offering hands-on lab exercises on offensive techniques. The fact of adding ethical hacking to the curriculum raises a variety of ethical and legal issues. Some students will use the acquired offensive hands-on skills in inappropriate and sometimes illegal ways. There is a concern that teaching dangerous skills to immature and unqualified students may be socially irresponsible. In addition, the misuse of information security expertise is serious business and could result in criminal prosecution, bad publicity, personal injury, cyber bullying, and termination of educational programs, among numerous other negative outcomes. The study in Cook, Conti, and Raymond (2012) discusses in detail the problem of misuse of the information security skills and provides detail of real-world incidents involving students.

In addition, unmonitored offensive hands-on lab exercises may be a breach of the law. Also, schools and educators may be held liable for the actions of their students. Hence, students may threaten their careers, hurt others, and put their institution's entire information security program at risk.

It is often criticized that offensive methods should not be taught to students since they only increases the population of "malicious hackers." There are those who feel that teaching offensive techniques are unethical (Harris, 2004), since there is always a potential that some students will use the tools and techniques in an irresponsible manner. Therefore instructors should not take on the responsibility of teaching new hackers. Some educators are concerned that teaching dangerous skills to immature and unqualified students may be socially irresponsible. For a good discussion on the ethical and legal concerns regarding the teaching of offensive techniques in the academic environment, readers can refer to Caltagirone, Ortman, Melton, Manz, King, and Oman (2006) and to Livermore (2007).

We disagree with this line of argument. The trend towards penetration testing in corporate businesses shows that offensive techniques can be used to increase the level of security of an enterprise. Consequently, students trained in offensive techniques do not necessarily become malicious hackers, but rather become competent security professionals. The fact of not studying and applying the techniques, tactics, and methodologies of attackers would leave large gaps in the knowledge base of graduates (Brutus et al., 2010; Ledin, 2011). However, it is obvious that there is no guarantee that very few of the students who have been taught offensive techniques in schools will be hackers in the future and perform malicious hacking activities against systems and networks.

A survey of security faculty members has been conducted to determine their attitudes toward teaching ethical hacking and penetration testing in schools (Livermore, 2007). The survey results showed more than 71% of the security faculty members agreed that schools should be teaching ethical hacking. In addition, students with ethical hacking skills will be better prepared to work as security administrators with better chances of landing jobs than students without these skills (Arce & McGraw, 2004; Logan & Clarkson, 2005; Wulf, 2003).

Nowadays, many academic institutions are including ethical hacking and penetration testing in their information security and Computer Science programs. For examples, the University of North Carolina (UNC) at Charlotte's College of Computing and Informatics is offering a course called Vulnerability Assessment and System Assurance. Ethical hacking techniques, finding new exploits, discovering vulnerabilities, and penetrating network perimeters are among the topics covered by the course. The course is based on case studies of ethical hacking with a strong lab component. The University of Abertay Dundee in Scotland is also offering a course in ethical hacking called Ethical Hacking & Computer Security. The course provides detailed knowledge of electronic attacks and the methods that criminals use to gain access and exploit a system.

It is clear that there are a number of problems with teaching ethical hacking. However, there are also a number of steps that schools and educators can take to reduce their liabilities, to prevent student's misconduct, and to help students to not misbehave and be responsible, including:

1. Offensive techniques should not be offered exclusively, but in a context where the emphasis is on improving the defensive techniques by learning how offensive techniques are implemented.
2. Students should be aware of the legal implications and the ethics of offensive attacks and ethical hacking. Students should be regularly briefed on the ethical use of offensive tools and techniques. Instructors should deliberately and frequently provide context for students as to why they are learning dangerous material. The survey results in (Livermore, 2007) show that 75% of the security faculty members feel that their schools should offer an ethics course as part of the information security curriculum. A similar number of faculty feel that ethics should be part of every information security course in the curriculum.
3. Educators should communicate regularly the downsides of malicious actions to their students. Students risk expulsion, criminal prosecution, and could threaten the existence of their institution's information security program.

4. Schools should communicate to the students the boundary between appropriate and inappropriate behavior, including knowledge of local and international law.
5. To limit their liability, schools that offer offensive techniques as a part of their security curriculum should mandate student to sign a code of conduct during the course registration. The code of conduct should spell out the boundaries for student behavior and the consequences for unacceptable behavior.
6. Schools that construct computer labs for teaching offensive techniques in their information security programs must take precautions to ensure that their labs are isolated from all networks outside the classroom, to minimize the chances of accidental or intentional abuse.
7. Schools teaching offensive techniques should establish a process to screen students for criminal background, unstable behavior and malicious activities prior to admission to an information security program (Livermore, 2011). In addition, schools may require a written approval from the student's parents before conducting any screening for criminal background. Also, schools should not violate the local law regarding conducting criminal screening on individuals.
8. Schools can ask regularly the students enrolled in an information security program to provide the MAC addresses of their personal laptops and mobile devices. In case of an attack, these MAC addresses can help computer forensics specialists to identify the devices that initiated the attacks.

Schools that take the above steps improve the chances of having a successful and problem free information security programs that teach ethical hacking techniques.

Evaluation of Learning Outcomes and Student Satisfaction

This section discusses the effect of introducing the new attack hands-on labs on the achievement of the SECb455 course outcomes (COs). The SECB455 course has five COs as shown in Table 2. Since SECB455 is an advanced course in information security, the outcomes have been selected carefully to reflect the top three levels in Bloom's taxonomy of cognitive domain (analysis, synthesis, and evaluation). After creating the course outcomes, 12 course topics were identified and mapped to the course outcomes. Four assessment tools are also selected to assess the achievements of COs including quizzes, exams (midterm and final), lab reports, and term project.

Table 2: Mapping the course outcomes to Blooms Taxonomy

Outcome	Level of Bloom's Taxonomy
CO1: Identify the most common networks attacks	Analysis (4)
CO2: Analyze counter measures of network attacks	Analysis (4)
CO3: Perform security auditing and vulnerability assessment.	Evaluation (6)
CO 4: Create new attack signatures.	Synthesis (5)
CO 5: Integrate IDS/IPS sensors.	Synthesis (5)

To assess the course outcomes we follow the course assessment process adopted by our institution. A nominated course coordinator assembles a course committee that includes all the lecture and lab instructors teaching the course in a given semester. During the first week of the semester, the course committee meets to decide on the assessment tools that will be used to assess the COs. They also decide on the corrective actions that will be applied to address the recommendations from the previous assessment cycle. Throughout the semester, the course committee applies the assessment tools to collect assessment data. By the end of the semester, the collected assessment data are mapped to the COs. The achievement level of each CO is then calculated in terms of mean and standard deviation using (1) and (2).

$$\mu(CO_i) = \frac{\sum_t \mu_t \times n_t}{\sum_t n_t}, \quad (1)$$

$$\sigma(CO_i) = \sqrt{\frac{\sum_t \sigma_t^2 \times n_t}{\sum_t n_t}}, \quad (2)$$

where μ_t and σ_i denote respectively the normalized mean and standard deviation of the students' marks when assessment tool t is used, and n_t denotes the number of students. For example, if three quizzes and two final exam questions are used to assess CO_i, the normalized mean and standard deviation of the students' marks are calculated separately for each tool, then (1) and (2) are used to calculate the achievement level for CO_i. After calculating the achievement level for each CO, the course committee meets again to discuss the assessment results and decide on the needed recommendations to address any discovered shortcoming. To close the assessment cycle, the course committee also discuss the effectiveness of the corrective actions applied during the semesters on the new assessment results.

During the 2005/2006 and 2007/2008 academic years, students enrolled in SECB455 were not offered any attack hands-on lab. Only the theoretical part of the attacks was usually described during the lecture time. However, starting from fall 2008 the course committee decided to offer the proposed ethical hacking hands-on lab exercises as a corrective action to improve the COs achievement levels. Three quizzes are used to compare the achievement of the COs before and after the new attack hands-on labs are introduced. These quizzes are directly mapped to CO1, CO2, and CO5 shown in Table 2. The grades of the students in the three quizzes are measured, normalized, and then aggregated using (1) and (2) to calculate the achievement level of the three COs.

Assessment Results

Figure 19 shows the students average grades for the three quizzes which are to evaluate the students' comprehension of the three attacks. It clearly shows that starting from 08/09 academic the total average grade has started improving. This is mainly due to the fact that the offered hands-on lab exercises allowed students to better anatomize the attacks and assimilate the concepts learned from the lecture. The students have learned better with the exercises which had a positive effect on their performance. For example, in case of Quiz 2, introducing the lab exercises improved the average student grade by 11.2% from 0.7 to 0.79 and maintained the improvement for the following two academic years.

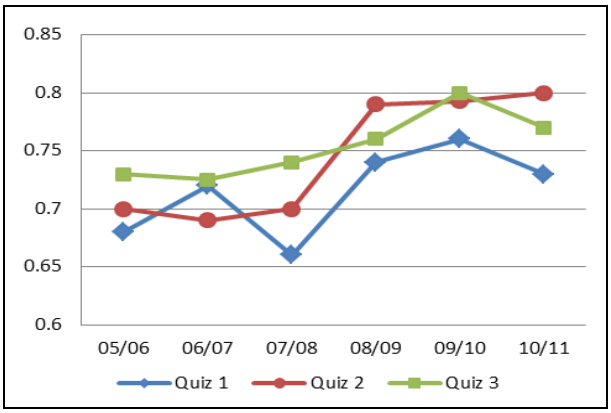

Figure 19: *Student performance in the three quizzes before and after introducing the DoS attacks labs*

Figure 20 illustrates the achievement of the three course outcomes for six consecutive years from 05/06 to 10/11. It shows a considerable improvement in the COs achievements level after introducing the attack hands-on labs. For example, the introduction of the proposed labs in 08/09 academic year improved the CO achievements level by 8.7%, 5.8%, and 6% for CO1, CO2, and CO5 respectively compared to the achievement levels in the year before.

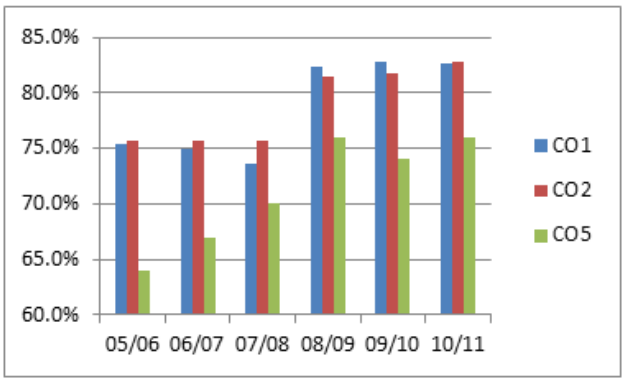

Figure 20: *Using assessment tools to map the new attack labs to the course outcomes*

Table 3 compares the CO achievement levels of the 06/07 and the 10/11 academic years in detail. It shows that the 9% improvement in the achievement level of CO1, from 75% to 82.3% (see Figure 20), can be interpreted as 14.9% and 5.8% increase in the number of students scored above 90% and 80% respectively, and 13.8% and 6.3% drop in the number of students scored below 60% and 70% respectively. Similar behaviour can also be observed for CO2 and CO5.

Table 3: Detailed description of the COs achievement

	CO-1		CO-2		CO-5	
	06/07	10/11	06/07	10/11	06/07	10/11
CO < 60%	20.8%	7.0%	14.8%	8.1%	25.9%	15.9%
60% ≤ CO < 70%	19.0%	12.7%	18.9%	12.8%	21.8%	18.2%
70% ≤ CO < 80%	21.9%	21.3%	24.4%	20.3%	22.5%	23.0%
80% ≤ CO < 90%	18.6%	24.5%	21.6%	23.1%	16.6%	20.8%
90% ≤ CO	19.7%	34.6%	20.4%	35.7%	13.2%	22.2%

Student's Satisfaction

Table 4 shows the results of an anonymous questionnaire that was administered to 110 students, who participated in the lab exercises, to measure their satisfaction level and collect their feedback regarding the discussed hands-on lab exercises. The results of the questionnaire showed that more than 85% of all students who answered the questionnaire believed the lab exercises to be useful and helped them better understand the underlying theoretical concepts associated with DoS attacks (Table 4). The questionnaire also revealed that 87% of the students were interested in similar exercises in other network security classes, and 86% would strongly recommend the lab exercise to other students.

Table 4: Student satisfaction questionnaire

Questions	Responses			
	Strongly Agree	Agree	Neutral	disagree
Did you enjoy the labs?	87%	10%	2%	1%
Do you think the labs are easy to follow and straightforward?	82%	10%	5%	3%
Do you feel you understand the DoS concepts better after performing the labs?	85%	13%	1%	1%
How likely are you to recommend the labs to others?	86%	11%	2%	1%
Would you like to see similar labs offered in your network security classes?	87%	8%	4%	1%
Laboratory exercises helped me to learn how to apply security principles and tools in practice.	85%	8%	5%	2%

Conclusion

It is necessary that students know how to attack and anatomize offensive techniques to truly understand how to defend networks and computer systems, and strengthen their security skills. This paper described offensive hands-on lab exercises on how to practically perform three common DoS attacks. The exercises are designed to be used as a part of an undergraduate-level course on network security and intrusion detection. The exercises allow students to better anatomize the DoS attacks in an isolated network laboratory environment.

A major ethical concern has been identified after analysing the alert logs generated by the IDS sensors installed in the university network segments. This is a dilemma when security students are exposed to hands-on lab exercises on offensive techniques. However, the ethical concerns of teaching students "hacking" are dwarfed by the need for knowledgeable, competent, and, above all, experienced computer security professionals in industry and government. Course assessment results also show a significant improvement in the achievement level of related course outcomes. In addition, the paper discussed a number of steps that schools teaching offensive techniques can take to reduce their liability, educate students about their ethical responsibilities, and prevent teaching dangerous skills to the wrong students.

Acknowledgment

The authors acknowledge the support of Emirates Foundation through Research Grant (2011/161).

References

Arce, I., & McGraw, G. (2004). Guest editors' introduction: Why attacking systems is a good idea. *IEEE Security & Privacy*, *2*(4), 17-19.

Arnett, K. P., & Schmidt, M. B. (2005). Busting the ghost in the machine. *Communications of the ACM-Spyware*, *48*(8), 92-95.

Bishop, M. (1997). The state of infosec education in academia: Present and future directions. *Proceedings of the National Colloquium on Information System Security Education*, 19–33.

Brutus, S., Shubina, A., & Locasto, M. (2010). Teaching principles of the hacker curriculum to undergraduates. *Proceedings of the 41st ACM Technical Symposium on Computer Science Education,* 122-126.

Burdach, M. (2010). *Hardening the TCP/IP stack to SYN attacks.* Retrieved from http://www.symantec.com/connect/articles/hardening-tcpip-stack-syn-attacks

Caltagirone, S., Ortman, P., Melton, S., Manz, D., King, K., & Oman P. (2006). Design and implementation of a multi-use attack-defend computer security lab. *Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, (9), 220c.

Cisco Catalyst 4500 Series Switches Configuration Guide. (2013), Retrieved from http://www.cisco.com

CommView Network Monitor and Analyzer. (2013). Retrieved from http://www.tamos.com

Cook, T., Conti, G., & Raymond, D. (2012). When good Ninjas turn bad: Preventing your students from becoming the threat. *Proceedings of the 16th Colloquium for Information System Security Education*, 61-67.

Dornseif, M., Gärtner, F. C., Holz, T., & Mink, M. (2005). *An offensive approach to teaching information Security: Aachen summer school applied IT security*. Technical report AIB-2005-02, RWTH Aachen.

Engage Packet Builder. (2013). Retrieved from http://www.engagesecurity.com

FrameIP Packet Generator. (2013). Retrieved from http://www.frameip.com

Frincke, D. (2003). Who watches the security educators? *IEEE Security and Privacy, 1*(3), 56–58.

Harris, J. (2004). Maintaining ethical standards for computer security curriculum. *Proceedings of the 1st Conference on Information Security Curriculum Development,* 46–48.

Hill, J. M., Carver, C. A., Jr., Humphries, J. W., & Pooch, U. W. (2001). Using an isolated network laboratory to teach advanced networks and security. *Proceedings of the 32nd SIGCSE Technical Symposium on Computer Science Education*, 36–40.

Juniper Networks SSG5 and SSG20 Secure Services Gateways. (2013). Retrieved from http://www.juniper.net/us/en/local/pdf/datasheets/1000176-en.pdf

Ledin, G. (2011). The growing harm of not teaching malware. *Communications of the ACM*, *54*(2), 32-34.

Livermore, J. (2007). What are faculty attitudes toward teaching ethical hacking and penetration testing? *Proceedings of the 11th Colloquium for Information Systems Security Education*, 111-116.

Livermore , J. (2011). Screening IA students for criminal background. *Proceedings of the 15th Colloquium for Information System Security Education,* 81-86.

Logan, P., & Clarkson, A. (2005). Teaching students to hack: Curriculum issues in information security. *Proceedings of the 36th SIGSE Technical Symposium on Computer Science Education,* 157-161.

Microsoft. (2011). *How to harden the TCP/IP stack against denial of service attacks in Windows Server 2003*. Retrieved from http://support.microsoft.com/kb/324270

Mink, M., & Freiling, F. C. (2006). Is attack better than defense? Teaching information security the right way. *Proceedings of the Conference on Information Security Curriculum Development,* 44-48.

Mullins, P., Wolfe, J., Fry, M., Wynters, E., Calhoun, W., Montante, R., & Oblitey, W. (2002). Panel on integrating security concepts into existing computer courses. *Proceedings of the 33rd SIGCSE Technical Symposium on Computing Education,* 365-366.

Radmin Advanced Port Scanner. (2013). Retrieved from http://www.radmin.com

Sharma, S. K., & Sefchek, J. (2007). Teaching information systems security courses: A hands-on approach. *Computers & Security Journal*, 26(4), 290-299.

Snort User's Manual. (2011). Retrieved from http://www.snort.org

Trabelsi, Z. (2011). Hands-on lab exercises implementation of DoS and MiM attacks using ARP cache poisoning. *Proceedings of the 2011 Information Security Curriculum Development Conference*, 74-83.

Vigna, G. (2003). Teaching hands-on network security: Testbeds and live exercises. *Journal of Information Warfare*, *2*(3), 8–24.

Xu, J., Zhang, J., Gadipalli, T., Yaun, X., & Yu H. (2011). Learning Snort rules by capturing intrusions in live network traffic replay. *Proceedings of the 15th Colloquium for Information Systems Security Education*, 145-150.

Yuan, D., & Zhong, J. (2008). A lab implementation of TCP SYN flood attack and defense. *Proceedings of the 9th ACM SIGITE Conference on Information Technology Education*, 57-58.

Wulf, T. (2003). Teaching ethics in undergraduate network security courses: The cautionary tale of Randal Schwartz. *Journal of Computing Sciences in Colleges, 19*(1), 90-93.

Biographies

Dr. Zouheir Trabelsi received his Ph.D. in Computer Science from Tokyo University of Technology and Agriculture, Japan, in 1994. From April 1994 until December 1998, he was a computer science researcher at the Central Research Laboratory of Hitachi in Tokyo, Japan. From November 2001 until October 2002, he was a visiting Assistant Professor at Pace University, New York, USA. In September 2005, he joined the College of Information Technology, United Arab Emirates University, where he is currently an Associate Professor of Information Security and the Master programs coordinator. Dr. Zouheir research interests include: Network security, Intrusion detection and prevention, Firewalls, TCP/IP covert channels, Information security education and curriculum development.

Dr. Walid Ibrahim received his Ph.D. in Systems & Computer Engineering from Carleton University (Ottawa, Canada) in 2002. In September 2004 he joined the Faculty of Information Technology, United Arab Emirates University, where he is currently the academic assessment coordinator and an Associate Professor with the Computer System Engineering track. Before joining the United Arab Emirates University, Dr. Ibrahim held several software R&D positions in worldwide leading telecommunication and semiconductor companies. Dr. Ibrahim research interests include: Reliability enabled EDA tools, ultra-low power designs, scalable and reliable interconnection topologies, VLSI testing and design for testability, applied optimization techniques.