Denial of Service Hands-on Lab for Information Assurance Education: A Case Study

Jianhua Yang, Yien Wang
TSYS School of Computer Science
Columbus State University
4225 University Ave.
Columbus, GA 31907 U.S.A.
yang_jianhua@ColumbusState.edu, wang_yien@ColumbusState.edu

ABSTRACT: *Conducting hands-on labs is playing a more and more important role on information assurance education. Nowadays, hands-on labs are essential for computer security classes. To help with lowering the cost of hardware and software and have the availability to offer offensive hands-on labs for online learning, we propose three DoS attack lab exercises which are different from the traditional DoS labs in terms of technique skills, hard cost, and online offering availability. Five years of offering these special designed DoS labs at Columbus State University shows that it can help students to reach their learning objectives and better assimilate the concepts covered in security classes.*

Keywords: DoS Attack, Network Security, Information Assurance, Cybersecurity Curriculum

Received:28 June 2015, Revised 29 July 2015, Accepted 16 August 2015

© 2015 DLINE. All Rights Reserved

1. Introduction

Information assurance education has become more and more important in recent years. Not only is it developed to an independent branch of computer science, but also injected into other disciplinaries of computing education. Today, most aspects of the national infrastructure depend upon the correct operation of computers and networks [1]. Computer network security class, as one important areas of information assurance education, also becomes extremely important as the whole society depends on the Internet to be secure to conduct many critical tasks [2][3][4][5]. To better understand the concepts covered in computer network security class, appropriately designed hands-on labs play a critical role.

In order to implement security solutions, students in computer network security class are normally required to conduct Denial of Service (DoS) attacking hands-on labs [6]. DoS attacking skills are fundamental topics in computer network security courses to teach students understanding hacking and intrusion techniques [7]. Obviously, these techniques are not defensive techniques. Although teaching defensive technique and designing related hands-on lab exercises in computer network security curriculum are popular and acceptable, the interest to teach offensive technique is growing rapidly in recent years. The primary reason is that many academic educators and industry practitioners insist on an idea that the best way to prepare a system defense is to understand the attacks that the system might face [7]. However, there are some concerns and disputes in teaching hacking techniques for college students. One typical concern is that teaching offensive skills to immature and beginning students may irresponsible [8]. Some educators claimed that one serious consequence of involving hacking skills and penetration testing in academic is to increase the population of "*malicious hackers*". A survey conducted in [7] clearly shows that as high as 85% of the students admitted that they had tried the learned DoS attacking techniques on outside beyond the isolated computer network laboratory environment. Web servers, email servers, file servers, and database servers were normally their primary targets. In addition, unmonitored offensive labs may be a breach of the law. Despite of the above concerns and disputes, one most striking point, which is supported by most academic educators and industry professionals, is that teaching ethical hacking, offensive techniques, and penetration skills in information assurance education curriculum can yield more high-quality security experts.

Traditional DoS attack hands-on lab exercises primarily include a) Ping of Death; b) Smurf Attacks; c) TCP (transmission control protocol) SYN (synchronization) flood; and d) Teardrop attacks. Ping of Death is an attack aiming to crash a target system by sending ping packets that exceed the maximum byte size of 65,536 bytes allowed by TCP/IP protocol. Smurf attack could generate significant network traffic on a target network by sending spoofed ping messages to a target computer host which correspondingly responds back by a broadcast address to flood the network. TCP SYN Flood attack can be launched by attackers through sending spoofed TCP SYN packets to initiate a TCP connection to a target host with its IP address as both source and destination IP. The SYN-ACK packets sent by the target host can be bounced back due to the identical source and destination IP address. Thus the SYN-ACK packets eventually overwhelm the target and cause the service stop working. Keeping pending half-open connection can consume target host memory. Spoofed IP address can quickly result in tons of pending TCP connections and finally exhaust target memory and bandwidth. Teardrop attack occurs at the time when fragmented IP packets are reassembled at destination host. Overlapped offset in different fragmented packets can cause a destination host to hang or crash when the destination host tries to reassemble the packets.

The above DoS hands-on lab exercises can definitely help students understand how to defend networks and computer systems, and strengthen their defensive and offensive skills. However, there are still some challenges and concerns that prevent us from adopting these labs. All the labs need some hardware and specific software. Even though most software needed, such as Snort, Frame IP Packet Generator, and Engage Packet builder, might be free of charge, most hardware are normally not free. For conducting the labs, each student needs at least one switch/router and three computers. If we support 10 students doing the labs concurrently, we would need 10 switches/routers-and 30 computers. This cost could tight most teaching-focused mid-size universities' budget. For safety and security reason, the LAN on which the labs are conducted must be isolated from the Internet. An isolated LAN cannot be accessed by online students. In Columbus State University (CSU), we have a large group of online students in Applied Computer Science Master Program: information assurance track. Therefore, the traditional DoS attack hands-on lab exercises are not suitable to CSU information assurance track online students.

In order to make our students not to miss the opportunity to practice ethical hacking techniques and be well prepared to become a qualified security professional, the solution we propose in CSU information assurance curriculum is to design some new DoS attack hands-on lab exercises which aim at exhausting computer system resources, such as hard disk capacity, memory, and CPU of a computer system. A denial of service occurs whenever any legitimate user of some services is prevented from using that service. All these three labs focus on prevention of local system resource exhaustion. The attacking side of the first lab is to exhaust a hard disk capacity resource in a computer system by running a shell script program. The defending part of this lab is to limit the capability of a specific user to use the hard disk resource. Similarly, other two labs focus on exhausting the memory and CPU of a computer system, respectively.

These labs can be conducted over a virtual computer environment which leads to a minimum cost and can be accessed by online students. With the labs, students can be provided with the required hands-on experience to improve their comprehensive knowledge of different DoS attacks, as well as having an in-depth understanding on the attacks and assimilating the concepts learned in security class, and gain insight of offensive techniques.

The paper is organized as follows. Section 2 presents the software and tools needed for implementing the three labs. Section 3 describes the details of new designed DoS attacking hands-on lab exercises. Some comments and feedback from students are given in Section 4. In Section 5, we conclude the paper, and discuss about future work.

2. Software and Tools

Unlike the traditional DoS attack, the labs we design for CSU computer network security class do not need an host and isolated local area network. Considering the other labs designed for this class, we let our students access a virtual lab system which is built with Oracle VirtualBox. In this system, we have four virtual machines set up in each one having a different OS installed. One of the virtual machines is installed with Linux Fedora 8. The other three virtual machines are installed with BT5, windows XP, and Kali, respectively. All the software used in this system and for all the labs are free of charge. The virtual system can be either installed locally on students' side, or accessed through VPN to a centralized system located inside of CSU. Students can set up their own lab environment on their own computer system with all the software we provide. In this situation, students can finish their labs locally without worryig about network traffic. But we found that hosting virtual system needs a powerful computer with at least 8G-memory. Some students may not have such high-end computer system due to a high cost. If so, the students need to access the system provided by CSU through VPN. But this incurs a network traffic issue if students do not have broad-band network at their home or office.

In terms of doing the DoS attack lab exercises, it does not need any other special software. Students are only required to understand Linux operation and shell programming. We ask students using the virtual machine with Fedora-8 OS to conduct the three labs. This can save budget and be suitable for mid-size teaching-focused universities. More importantly, this can be available for online students. It can also reach the objective of the hands-lab exercises: havuing an in-depth understanding of offensive techniques.

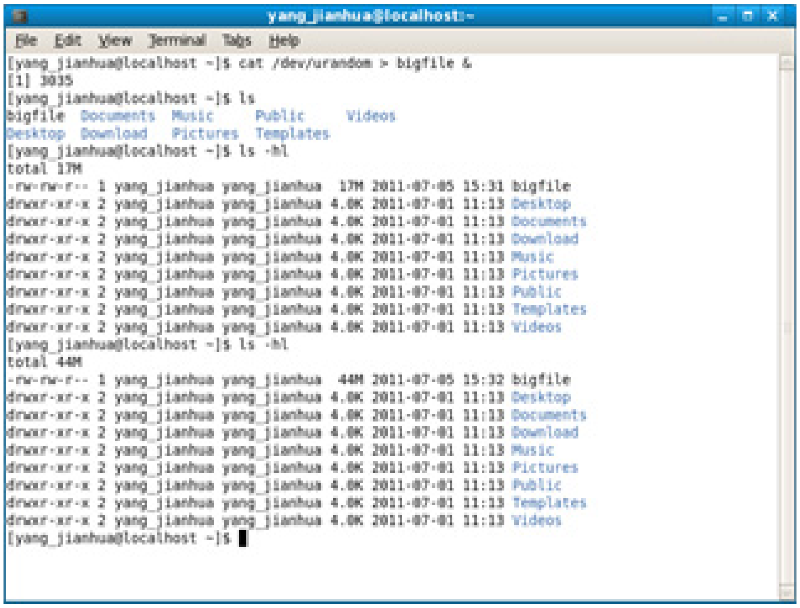
3. Denial of Service Labs

After checking the four traditional DoS attacks, we found that the basic idea behind DoS attack is to crash a computer system through exhausting the system resources. We bring the same idea to the new designed DoS attack lab exercises. A DoS attack could occur due to a third party malicious packet attack or exhausting your system resources through an unlimited running program.

3.1 Hard Disk Exhaustion Attack

In the first lab, we use a small shell code to exhaust a hard drive available space via increasing the size of a file unlimitedly. Students are required to propose an approach to prevent the hard drive space exhausting attack from happening.

In order to start this lab, students need to log into Fedora 8 system as an unprivileged user. We assume that this user is referred to as yang_jianhua throughout this lab. A terminal is open at Fedora 8, and a Linux command "*cat /dev/urandom >bigfile &*" is executed. This command indicates that random numbers are generated and write to a file with name "*bigfile*" continuously. The size of this file is increased continuously due to execution of this command. The file size increment can be verified by executing the command "*ls – hl*" as shown in Figure 1.

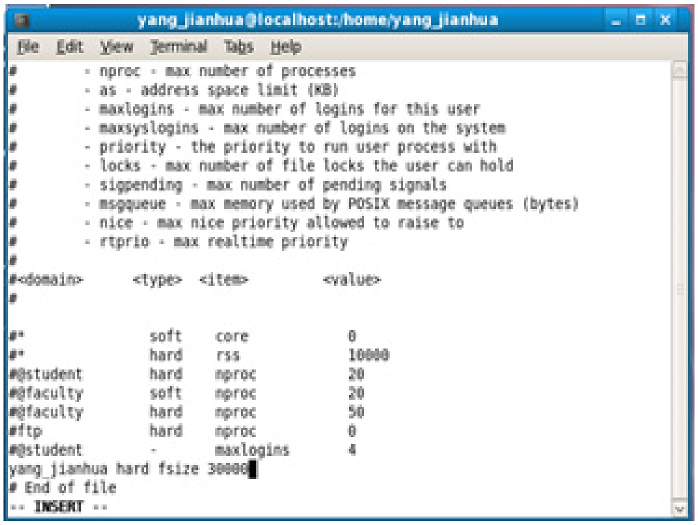


```
[yang_jianhua@localhost ~]$ cat /dev/urandom > bigfile &
[1] 3035
[yang_jianhua@localhost ~]$ ls
bigfile  Documents  Music     Public     Videos
Desktop  Download   Pictures  Templates
[yang_jianhua@localhost ~]$ ls -hl
total 17M
-rw-rw-r-- 1 yang_jianhua yang_jianhua  17M 2011-07-05 15:31 bigfile
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Desktop
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Documents
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Download
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Music
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Pictures
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Public
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Templates
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Videos
[yang_jianhua@localhost ~]$ ls -hl
total 44M
-rw-rw-r-- 1 yang_jianhua yang_jianhua  44M 2011-07-05 15:32 bigfile
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Desktop
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Documents
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Download
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Music
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Pictures
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Public
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Templates
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Videos
[yang_jianhua@localhost ~]$ █
```

Figure 1. Verify the file size change

As shown in Figure 1, it displays that the size of "*bigfile*" has been increased from 17M to 44M due to running the shell script "*cat /dev/urandom > bigfile &*". If we did not limit the size of "*bigfile*", it would eventually increase the size of the file as large as occupying the entire existing available space of the hard disk, and then crash the whole computer system.

The way to defeat such kind of attack is to limit the size of a file created by a user. Adding some rules to Fedora 8 system configuration file "limits.conf" can prevent the size of a file from constant growing. For example, if a rule "*yang_jianhua hard fsize 30000*" is added to user "*yang_jianhua*", this attack can be prevented. This rule indicates that the size of a file created by user "*yang_jianhua*" is capped to 30M. In order to modify file "*limits.conf*", "*su root*" must be executed because user "*yang_jianhua*" logged in as an unprivileged user. The system must reboot to make the configuration change effective. Figure 2 shows how to change '*limits.conf*' file.

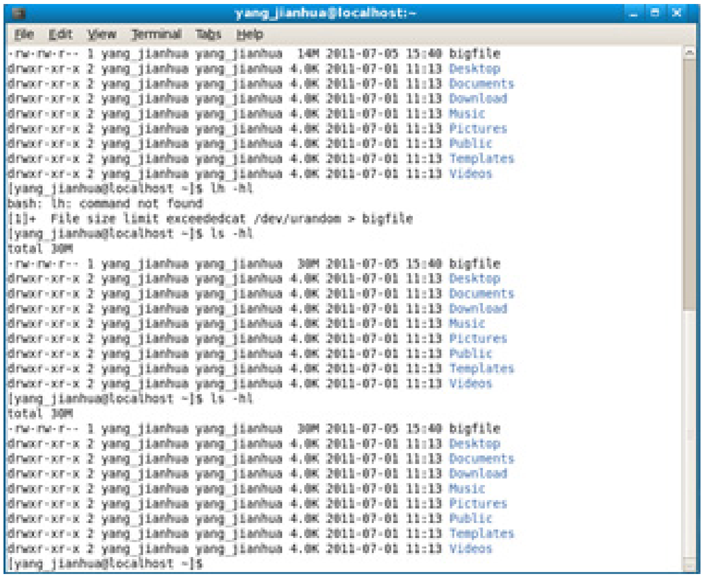

yang_jianhua@localhost:/home/yang_jianhua
File Edit View Terminal Tabs Help

```
#        - nproc - max number of processes
#        - as - address space limit (KB)
#        - maxlogins - max number of logins for this user
#        - maxsyslogins - max number of logins on the system
#        - priority - the priority to run user process with
#        - locks - max number of file locks the user can hold
#        - sigpending - max number of pending signals
#        - msgqueue - max memory used by POSIX message queues (bytes)
#        - nice - max nice priority allowed to raise to
#        - rtprio - max realtime priority
#
#<domain>      <type>  <item>         <value>
#

#*               soft    core            0
#*               hard    rss             10000
#@student        hard    nproc           20
#@faculty        soft    nproc           20
#@faculty        hard    nproc           50
#ftp             hard    nproc           0
#@student        -       maxlogins       4
yang_jianhua hard fsize 30000
# End of file
-- INSERT --
```

Figure 2. Edit limits.conf

After restarting the system, the rule to cap the file size for user "*yang_jianhua*" is effective. This can be justified by executing command "*cat /dev/urandom > bigfile &*" again. But at this time, you see a warning message "*File size limit exceeded*" shown up. We check the files for multiple times as shown in Figure 3 that the size of file "*bigfile*" is capped to 30M and stays. This lab shows that this type of DoS attack can be prevented from occurring through modifying configuration file "*limits.conf*".



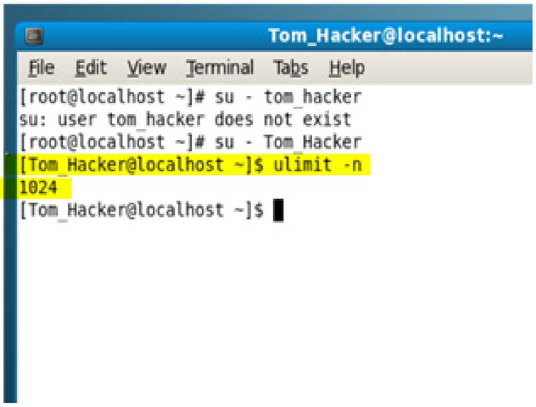
```
-rw-rw-r-- 1 yang_jianhua yang_jianhua  14M 2011-07-05 15:40 bigfile
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Desktop
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Documents
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Download
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Music
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Pictures
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Public
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Templates
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Videos
[yang_jianhua@localhost ~]$ lh -hl
bash: lh: command not found
[1]+  File size limit exceededcat /dev/urandom > bigfile
[yang_jianhua@localhost ~]$ ls -hl
total 30M
-rw-rw-r-- 1 yang_jianhua yang_jianhua  30M 2011-07-05 15:40 bigfile
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Desktop
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Documents
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Download
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Music
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Pictures
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Public
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Templates
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Videos
[yang_jianhua@localhost ~]$ ls -hl
total 30M
-rw-rw-r-- 1 yang_jianhua yang_jianhua  30M 2011-07-05 15:40 bigfile
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Desktop
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Documents
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Download
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Music
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Pictures
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Public
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Templates
drwxr-xr-x 2 yang_jianhua yang_jianhua 4.0K 2011-07-01 11:13 Videos
[yang_jianhua@localhost ~]$
```

Figure 3. Hard disk exhaustion attack

3.2 Memory Exhaustion Attack

When a file is open in a computer system, a FCB (File Control Block) is created to control the access of the file. The FCB is stored in the memory of a computer system which is managed by computer operating system, such as Fedora 8. Each FCB has a certain size, and is kept in memory until the open file is closed. The more FCBs a computer system has created, the more memory of the computer system is consumed. If all the open files in a computer system are not closed and more and more new files are open, more and more FCBs would be created. Thus, more and more computer system memory would be consumed by more and more FCBs. Finally the computer system memory would be exhausted up since each computer system has a fixed size of memory.

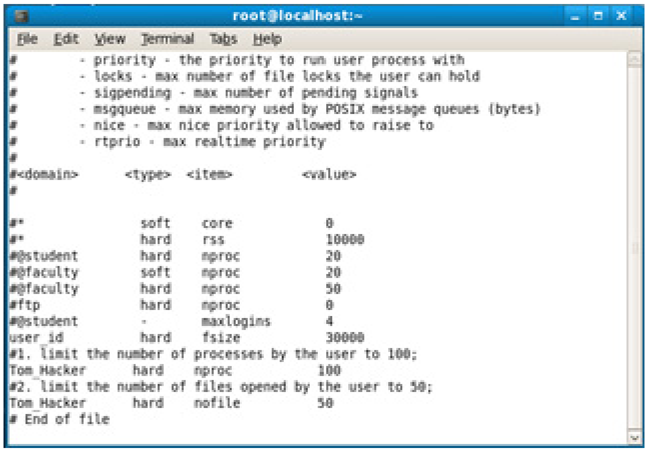
Memory exhaustion attack is to use the above idea to crash a computer system. For conducting this attack, a user needs to login to Fedora 8 as an unprivileged user such as "*Tom_Hacker*". We use command "*ulimit –n*" to check the default maximum number of files allowed to open in this system. For our system, this number was 1024 as shown in Figure 4.



```
[root@localhost ~]# su - tom_hacker
su: user tom_hacker does not exist
[root@localhost ~]# su - Tom_Hacker
[Tom_Hacker@localhost ~]$ ulimit -n
1024
[Tom_Hacker@localhost ~]$
```

Figure 4. Check number of files allowed to open

The maximum number of files allowed to open for a user at Fedora 8 can be changed through modifying system configuration file "*limits.conf*". If limiting user "*Tom_Hacker*" to the maximum number of files opened to be 50, rule '*Tom_Hacker hard nofile 50*" needs to be added to configuration file "*limits.conf*" as shown in Figure 5.

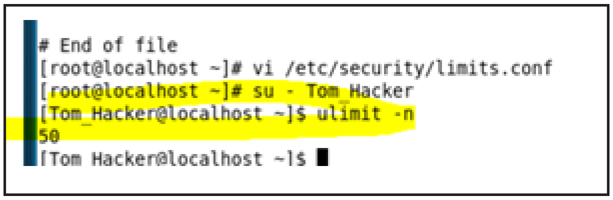


```
#        - priority - the priority to run user process with
#        - locks - max number of file locks the user can hold
#        - sigpending - max number of pending signals
#        - msgqueue - max memory used by POSIX message queues (bytes)
#        - nice - max nice priority allowed to raise to
#        - rtprio - max realtime priority
#
#<domain>      <type>  <item>         <value>
#

#*               soft    core            0
#*               hard    rss             10000
#@student        hard    nproc           20
#@faculty        soft    nproc           20
#@faculty        hard    nproc           50
#ftp             hard    nproc           0
#@student        -       maxlogins       4
user_id          hard    fsize           30000
#1. limit the number of processes by the user to 100;
Tom_Hacker      hard    nproc           100
#2. limit the number of files opened by the user to 50;
Tom_Hacker      hard    nofile          50
# End of file
```

Figure 5. Change configuration file

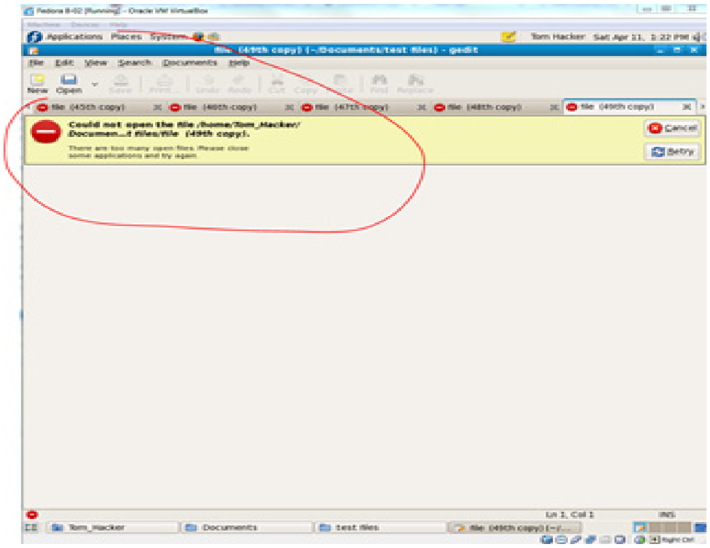
After changing the configuration file and restarting the computer system to make the configuration change effective, command "*ulimit –n*" is executed again to check the maximum number of files allowed to open, now it is changed to 50 instead of 1024, as shown in Figure 6.



```
# End of file
[root@localhost ~]# vi /etc/security/limits.conf
[root@localhost ~]# su - Tom_Hacker
[Tom_Hacker@localhost ~]$ ulimit -n
50
[Tom_Hacker@localhost ~]$ █
```

Figure 6. Recheck the number of files allowed to open

Instead of making a shell code, such as "*find . -name '*'-exec gedit {} +*", to open the same file 50 times or 50 different files, An alternative is that we can simply open the same file for 50 time manually. We eventually found "Could not open the file…" error message generated by the system at the 50th time to open a file. This message can be observed from Figure 7.


Fedora 8-02 [Running] - Oracle VM VirtualBox
Applications Places System
Tom Hacker Sat Apr 11, 1:22 PM
file (49th copy) (~/Documents/test files) - gedit
File Edit View Search Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Find Replace
file (45th copy)
file (46th copy)
file (47th copy)
file (48th copy)
file (49th copy)
Cancel
Retry
Ln 1, Col 1
INS
Tom_Hacker
Documents
test files
file (49th copy) (~/...

Could not open the file /home/Tom_Hacker/Documen...t files/file (49th copy).

There are too many open files. Please close some applications and try again.

Figure 7. Memory exhaustion attack prevention

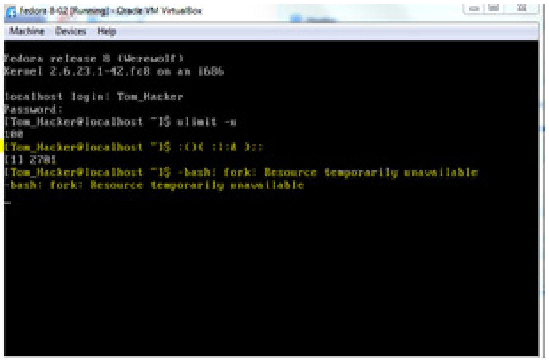
3.3 CPU Exhaustion Attack

One way to attack CPU is to make CPU busy and then make it unable to process any further requests. Continuously new created processes can keep CPU busy and eventually exhaust all the CPU time. A new process can be created using "*fork()*" function under Linux OS, such as Fedora 8. As long as a new process is forked, OS will create a PCB (process control block) to control and access the process. Each PCB needs to consume memory space, and each new forked process needs to be loaded into memory. Some processes may have data area which needs more memory space as well. Forked processes wait for CPU processing in a waiting queue. If the CPU is ready to execute the processes, they are brought into a ready queue. Thus, forking processes continuously not only exhausts CPU time, but also exhausts computer system memory.

In this attack, we first do not limit the number of processes allowed by user "*Tom_Hacker*", and execute fork bomb *":(){ :|:& };:"* code. We found the system crashed because of the unlimited number of processes forked. Restart the system, and modify configuration file "limits.conf" by adding a rule "*Tom_Hacker hard nporc 100*" to the file as shown in Figure 5. Reboot the system to make the change effective. We run the forked bomb code again, and it shows that the system is not crashed since as long as 100 processes are forked, the system stopped forking any more processes as it is shown in Figure 8a and Figure 8b.

4. Comments and Feedback

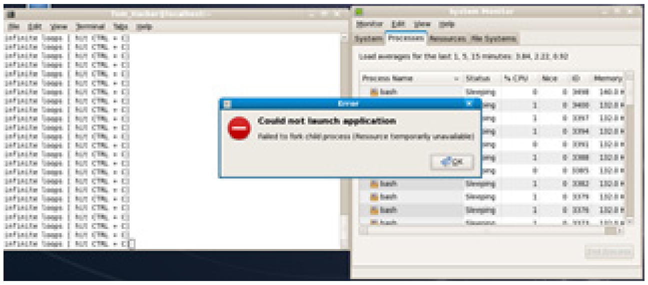
As we know that even though there are some concerns about teaching offensive technique, many academic institutions have included ethical hacking and penetration testing in their information assurance curriculum. We have used the DoS attack lab exercises in CSU computer network security class for five years. The survey from each year shows that more than 90% of the students are satisfied with DoS attack lab exercises. Every year, our students return excellent comments and feedback to us. We summarize them as the following.



```
Fedora release 8 (Werewolf)
Kernel 2.6.23.1-42.fc8 on an i686

localhost login: Tom_Hacker
Password:
[Tom_Hacker@localhost ~]$ ulimit -u
100
[Tom_Hacker@localhost ~]$ :(){ :|:& };:
[1] 2701
[Tom_Hacker@localhost ~]$ -bash: fork: Resource temporarily unavailable
-bash: fork: Resource temporarily unavailable
_
```

Figure 8a. Process forking and prevention


File Edit View Terminal Tabs Help
Monitor Edit View Help
System Processes Resources File Systems
Process Name Status % CPU Nice ID Memory
Error
OK

```
infinite loops | hit CTRL + C
infinite loops | hit CTRL + C
infinite loops | hit CTRL + C
infinite loops | hit CTRL + C
infinite loops | hit CTRL + C
infinite loops | hit CTRL + C
infinite loops | hit CTRL + C
infinite loops | hit CTRL + C
infinite loops | hit CTRL + C
infinite loops | hit CTRL + C
infinite loops | hit CTRL + C
infinite loops | hit CTRL + C
infinite loops | hit CTRL + C
infinite loops | hit CTRL + C
infinite loops | hit CTRL + C
infinite loops | hit CTRL + C
infinite loops | hit CTRL + C
infinite loops | hit CTRL + C
infinite loops | hit CTRL + C
infinite loops | hit CTRL + C
infinite loops | hit CTRL + C
infinite loops | hit CTRL + C
infinite loops | hit CTRL + C
infinite loops | hit CTRL + C
```

Load averages for the last 1, 5, 15 minutes: 3.84, 2.22, 0.92

Could not launch application

Failed to fork child process (Resource temporarily unavailable)

Figure 8b. Forking error message

1) The labs can help them to understand the mechanism behind DoS attacks.

2) Ethics and Laws class should be added to information assurance education curriculum. At least network securityclass needs to mention the US laws and the laws in different states about hacking.

3) Some educators suggest criminal background check to students prior to admission to information assurance track [9]. But most students are against the idea.

4) Students prefer to use a local virtual system than accessing a remote one through VPN.

5) Every year at the beginning of the semester to offer computer security class, we mandate our students to sign on an agreement and mail the agreement back to the class instructor. Students suggest that signing a code of conduct can be done online.

6) Students suggest that for each offensive lab, a lecture briefed on the ethical use of offensive tools, and the downsides of malicious actions should also be offered.

7) Students prefer to know why some hacking techniques and dangerous materials are covered in information assurance education.

5. Conclusion and Future Work

Defensive and offensive techniques are both presented in information assurance education. Along with defensive techniques to be required, offensive hacking techniques are also an important component of cybersecurity and computer science program at most academic institutions. Security hands-on labs play a significant role in terms of helping students to assimilate the concepts and ideas covered in cybersecurity class. Any hands-on lab offered in an institution must balance between budget, possibility, availability, and the consequences, especially for offensive security lab exercises. Considering the situation in CSU, we do not have enough budgets to support purchasing expensive hardware and software for penetration test solely, yet there is a necessity to provide labs for online students. Therefore, we propose specially designed DoS lab exercises in this paper: hard disk exhaustion attack; memory exhaustion attack; and CPU exhaustion attack. For all of these labs, students only need a Fedora system and write a small piece of Linux/UNIX shell code. The most important point is that the labs can help our students to reach the learning objectives, anatomize the attacks, assimilate the concepts learned from the lecture, and be prepared to become a security professional. In the future, considering the students' feedback, we will add ethics and laws section in computer network security class.

References

[1] Hill, J.M.D., Curtis, J., Carver, A., Humphries, J.W., Pooch, U.W. (2001). Using an Isolated Network Laboratory to Teach Advanced Networks and Security. *In :* Proceedings of the 32nd SIGCSE Technical Symposium on Computer Science Education, 36-40. ACM Press.

[2] Amoroso, E. (1994). Fundamentals of Computer Security Technology. Prentice Hall.

[3] Amoroso, E. (1999). Intrusion Detection: An Introduction to Internet Surveillance, Correlations, Trace Back, Traps, and Response. Intrusion.Net Books.

[4] Bishop, M. (2002). Computer Security-Art and Science. Boston, MA: Pearson Education, Inc.

[5] Northcutt, S., Novak, J. (2002). Network Intrusion Detection. 3rd ed., New Riders.

[6] 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. *In* :Proc. of the 39th Annual Hawaii International Conference on System Sciences, (9), 220c. Apr. il.

[7] Trabelsi, Z., Ibrahim, W. (2013). A Hands-on Approach for Teaching Denial of Service Attacks: a Case Study. J*ournal of Information Technology Education: Innovations in Proactive*, 12, 299-319.

[8] Cook, T., Conti, G., Raymond, D. (2012). When Good Ninjas Turn Bad: Preventing Your Students from Becoming the Threat. In Proc. of the 16th Colloquium for Information System Security Education, 61-67. August..

[9] Livemore, J. (2007). What Are Faculty Attitudes toward Teaching Ethical Hacking and Penetration Testing? *In*: Proceedings of 11th Colloquium for Information System Security Education, p. 81-86. September.

7. Author biographies

Jianhua Yang earned his Ph.D. degree in computer science from University of Houston, Houston, TX USA at 2006. He is currently working at TSYS School of Computer Science, Columbus State University (CSU), Columbus, GA USA as a Full Professor. Before joining CSU, he was an Assistant Professor at Bennett College from 2006 to 2008, University of Maryland Eastern Shore from 2008 to 2009, and Associate Professor at Beijing Institute of Petro-Chemical Technology, Beijing, China from 1990 to 2000. His current research interests are computer network and information security.

Yien Wang, a member of ACM and STARS, is currently a Graduate Assistant at TSYS School of Computer Science, Columbus State University. Her research interest is in cybersecurity and she is currently conducting research on intrusion detection under Dr. Jianhua Yang while pursuing her Master of Science in Applied Computer Science degree with concentration in Information Assurance.