Teaching Cybersecurity at the "Seams"

Steven Rigby, *University of Maryland University College*, and Amjad Ali, *University of Maryland University College*

Abstract – ***Many computer security programs supplement their courses by providing labs to fortify concepts being taught, however, often these labs are taught in isolation and do not allow students to see the complexity of integrating a systems of systems architecture. The "seams" of these security systems are where deep learning happens and where attacks slip through. This paper discusses a capstone course designed to help students integrate security systems with all of its interconnecting parts and see the importance of putting these pieces together securely.***

I. Introduction

The need for cyber-security professionals continues to grow at an astonishing rate as the number of threats and risks to our nation's computer systems increase. Each year we are witnessing more complicated attacks at every level of the enterprise infrastructure while attack tools have become automated and distributed freely on the Internet. In 2005, the President's Information Technology Advisory Committee made up of leading experts in academia and industry published a report entitled "Cyber Security: A Crises of Prioritization" [1] in which they state:

the Nation's cyber security problems have been building for many years and will plague us for many years to come. They derive from a decades-long failure to develop the security protocols and practices needed to protect the Nation's IT infrastructure, and to adequately train and grow the numbers of experts needed to employ those mechanisms effectively.

Academia has responded to the growing demand for cyber-security professionals by offering new undergraduate and graduate programs focused on cybersecurity. The success of these students will largely be based on the ability to understand and manage complexity, ambiguity, and solve ill-structured problems. One way to help students learn how to integrate disparate security systems is to provide an environment where students can create and integrate systems that work together to provide security to an organization. Very rarely are students provided an environment where they see these systems come together in its entirety. Previously, this type of environment would be cost prohibitive, however with the combination of virtual hardware, and software, students can now simulate real world environments.

Virtual software, hardware, and servers have helped create new and exciting possibilities for creating cybersecurity environments that can mimic real-world environments [2][3][4][5][6][7]. These virtual environments provide an instructional tool for allowing students to look at the "seams" of cybersecurity systems and develop the skills necessary to make an immediate contribution to future employers. Understanding how these interconnecting parts work as a whole to protect systems provides a solid foundation for students to take into the workforce.

A capstone cybersecurity course for computer information technology students was designed to help students see the complexity of integrating many of the necessary parts that make up a cybersecurity system.

The capstone course includes eight real-world projects that successively build upon and integrate with each other. Students are expected to research how to accomplish each project with limited instruction given in class. The instructor's role is one of directing and facilitating as well as helping students when they run into roadblocks.

This 14-week course has been evolving over the last couple of years and can support nine teams of four students. Each team has the necessary environment to accomplish all of the phases in the course. Due to the complexity and higher level thinking required for this course, students will have already taken courses in operating systems, networking, programing, and security.

The equipment used for this course includes two VMware ESX servers, one Cisco 5520 firewall for creating security contexts or "virtual firewalls" and nine Cisco 5505 ASA firewalls for IPSEC VPNs. Figure 1 shows the physical layout of the equipment used for this course. Figure 2 shows the logical view of all of the interconnected parts that make up this course. The course is made up of 8 phases or projects with a final project that help students see how cybersecurity systems interconnect and develop

ISBN 1-933510-96-X/$15.00

the skills to install, configure, troubleshoot, and maintain these systems.

II. Phase 1

Phase one includes helping students understanding how to secure a perimeter network. A Cisco ASA firewall with the ability to create "virtual" firewalls is used to facilitate this. Each team of four students is provided a firewall context or "virtual firewall" that was connected to the Internet to install and configure.

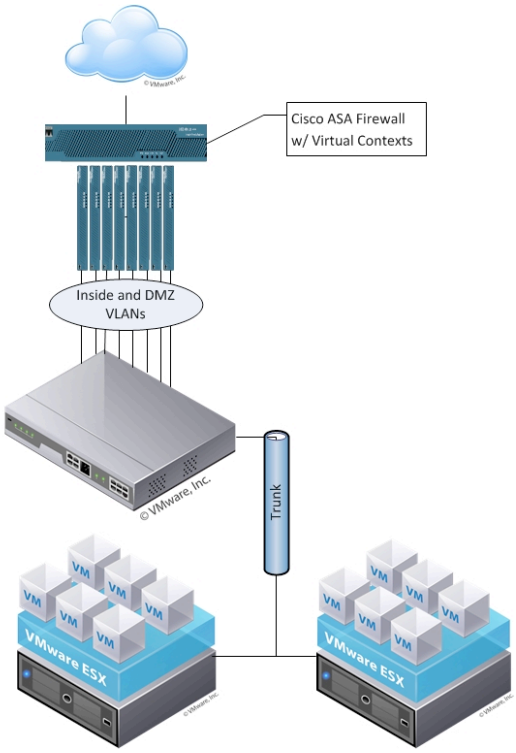

Figure 1 Physical View.

Teams are required to research and then configure the perimeter firewall with banners, timeouts, SSH access, Network Access Translation (NAT), Port Address Translation (PAT), AAA authentication, and ASDM access. Each team also creates outside, dmz, and inside networks with appropriate security levels and VLANS. Additionally, students learn how to install and configure virtual machines (VMs) on VMware ESX servers. This also involves students learning how to create "virtual switches" so that VMs are put on the correct VLans in order to talk to the firewall.

This phase of the project helps students visualize how a dmz and inside networks communicate through the firewall as well as understanding security levels and access control lists. Students accomplish this phase during the first two weeks of the course.

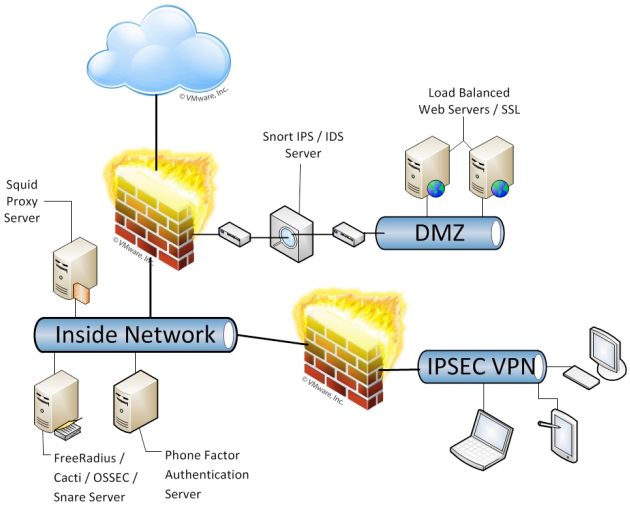

Figure 2 Logical View.

III. Phase 2

Phase two of the course allows teams to see how traffic from the Internet is translated through the firewall to a webserver using SSL/TLS. Additionally, teams are required to install/configure load balancing on two webservers to provide redundancy. Most organizations utilize some type of load balancing for incoming connections and this helps students to see how networks can be designed to provide high service level agreements. It is left up to the teams to decide which RFC 1918 address space they will use for the dmz and inside networks.

In order to accomplish this phase, teams install, configure, and harden two Windows 2008 R2 VM's servers with IIS. Students then configure Microsoft load balancing so that both websites will respond to one virtual IP address. After setting up the webservers and load balancing, the students configure the firewall to translate a public outside IP address to the private dmz load-balanced IP address. This involves creating a static NAT translation on the firewall as well as access-lists to allow http and https traffic from the Internet to the outward facing IP address that is being translated to the dmz virtual IP.

Through this phase, students begin to understand how firewalls are used to translate traffic from the public Internet to private networks securely. They also begin to understand how statefull packet inspection protects networks on higher security levels and how access-control lists create "holes" into networks and the need to limit the

ISBN 1-933510-96-X/$15.00

ports allowed in. Teams are required to accomplish this phase during week 3 of the course.

IV. PHASE 3

During this phase, teams learn how to use 802.1x authentication to provide secure remote access into organizations and how remote access integrates into the overall security design.

This phase requires teams to set up an IPsec VPN connection from the Internet to their second cisco ASA firewall that authenticates users through Phonefactor and FreeRadius. This allows VPN users to access the inside network from the Internet through a secure encrypted tunnel. Phonefactor is a software program that provides multifactor authentication by calling the remote vpn user on their cell phones as part of the authentication process.

The authentication process includes teams setting up an IPsec VPN profile on the Cisco ASA that points to Phonefactor as the Radius server. Phonefactor does not authenticate the reqest; rather it sends the request to FreeRadius for authentication. If the authentication request to FreeRadius is successful, Phonefactor will call the vpn user on their cell phone. After the vpn user answers the call and enters the correct key or password, Phonefactor will then respond back to the firewall (cisco ASA) that the user successfully authenticated. Finally, the vpn user will be given an IP address from a specific pool created on the ASA that allows the vpn users to access the inside network.

Another requirement of phase 3 is for teams to set up a cacti VM to receive SNMP traps from both cisco ASA's. This involves configuring the firewalls to set up and secure SNMP using a community name and password and specifying the SNMP server's IP address. The cacti server receives these messages and can create interface statistics on the bandwidth utilization of each interface on the firewalls.

This phase provides some excellent learning opportunities for learners to understand how authentication requests happen before receiving an IP address onto the internal network using 802.1x authentication. Additionally, teams learn how to configure split-tunneling so that compromised vpn-users are not allowed simultaneous connections to the Internet and internal networks. Students may have heard about 802.1x authentication protocols; however, until they have installed, configured, and successfully implemented this authentication protocol, they will not have the conceptual understanding of how these security systems work together.

Many students mention that this is the "coolest" thing they have done in their academic studies, which suggests that although this project may be difficult and frustrating at times, it is also rewarding when completed. Because of the increased time needed to accomplish the many aspects of this phase they are allowed week 4 and week 5 to accomplish these activities.

V. PHASE 4

After accomplishing the necessary requirements for secure remote access in phase 3, teams begin to look at how to protect users on the network from social engineering attacks and malware. The requirements for this phase are for teams to set up a SQUID proxy server VM that is configured to perform whitelisting and blacklisting.

Whitelisting is configuring a list of approved websites that users are allowed to access. Configuring the browser to use a proxy server on a specific port will redirect web requests to the proxy server. These changes are usually located within the "LAN Settings" section of the web browser. When a user opens up their web browser, the browser will ask for a username and password. After entering the whitelist username and password the proxy will only allow the user to go to the websites that have been approved. When a user enters a URL that is not on the approved list, a page is displayed stating that the site is not allowed and to email their system administrator if this site is needed for work purposes. To ensure that these settings are not bypassed, an entry in the firewall that only allows the proxy server to make outbound requests for the user network is used.

Whitelisting is the most effective use of proxy servers due to the inability for users to browse to sites that are infected. Even if the user brings in an infected USB drive that infects the computer with spyware, the spyware will not be able to communicate back to its owner or handler. Keyloggers, spyware, trojan horse, cross site scripting XSS, and most other types of malware would be disrupted by implementing whitelists using proxy servers.

Blacklisting is different from whitelisting in that to users are not restricted to only sites that have been approved. Blacklisted users that login through a proxy are allowed to go to any site that is not restricted by the blacklist of unapproved sites. These blacklists can be downloaded from companies that keep a list of categorized sites up to date. For example, you can download blacklists that are categorized so that if you don't want users to go to gambling websites during work, you can tell the proxy server that gambling sites are prohibited.

After teams install SQUID they need to configure the service to prompt users for a username and password.

ISBN 1-933510-96-X/$15.00

Based upon the username entered, the service will either whitelist or blacklist the user. Teams are also required to create a firewall rule that only allows the proxy server access to the Internet so that users do not bypass the proxy settings. Additionally, teams need to create a bash script that will download and install blacklists from the Internet. This phase is accomplished during week 6 of the semester.

VI. Phase 5

At this point in the course teams will have accomplished implementing security systems and seeing how they are interconnected and interrelate. The focus for the next few phases of the course turns to intrusion detection using security information event management (SIEM). SIEM technology is used to monitor systems for security alerts from servers and network equipment. This phase includes students learning how to install and configure an open source SIEM product called OSSEC.

OSSEC is an open source host-based intrusion detection system (HIDS) that performs monitoring of most operating systems. Some of the features of OSSEC are integrity checking, rootkit detection, log analysis, windows registry monitoring, and active response. For this phase, students are required to install and configure the OSSEC server software on a Linux VM on the inside network. Teams will then create agent accounts on the OSSEC server that can be applied to the two windows servers on the DMZ. After creating these accounts, the teams will install the client agent on the two webservers. Monitoring will take place by configuring the OSSEC web GUI to report on any suspect activities on the webservers in the DMZ.

In order to successfully complete this phase, students will need to understand how to create communication from the dmz, which is on a lower security level to the inside network, which is on a higher security level. By default, no communications are allowed to a higher security level network. This requires teams to create another static NAT translation with appropriate access-control lists on the firewall. During this phase, students begin to understand what is required to secure communications from less secure networks like the DMZ to more secure networks. Intruders that compromise a server on the DMZ will begin the process of scanning for servers on internal networks. Having these connections as limited as possible makes it more difficult for intruders. Teams are required to accomplish these tasks during week 7.

VII. Phase 6

The next phase involves helping teams understand how network-based intrusion detections systems (NIDS) work within a cybersecurity system. During this phase, teams are required to install and configure a Snort VM on the inside network as NIDS. To accomplish this, teams will install one virtual interface on the DMZ with another on the inside network. The interface on the DMZ will be used to listen for vulnerabilities on the DMZ network while the one on the inside will be used for management.

Students will spend time learning how Snort works and configure and tune the appropriate Snort pre-processors to detect malicious activity on the network. Teams will also be required to sign up to receive rule files from Sourcefire to integrate within their Snort VM. These rules will allow students to actually see attacks coming into their DMZ network. Snort also has a web GUI to monitor and view alerts. To verify that their snort VM's are working, teams perform vulnerability scans on their webservers.

One of the powerful features of Snort is that custom rules can be created to alert for specific traffic. Another requirement of this phase is for students to create a custom rule to alert on any traffic with a specific word contained in the packet. The rule should alert even if there are blank spaces between characters and either upper or lower case. In order to accomplish this, students will learn how to use regular expressions in their rules they create.

Being able to create custom rules using regular expressions is essential for practicing security professionals. There are times when no NIDS rule exists for the type of packet that needs to be alerted and students that understand how to alert on different types of traffic will be more effective in watching for intruders. Due to the time required to learn Snort and regular expressions, teams work on this phase during weeks 8 and 9.

VIII. Phase 7

After configuring Snort as a NIDS students learn how to use Snort as a network intrusion prevention system (NIPS). During the previous phase, malicious traffic was only logged and alerted. The packet was able to arrive at its final destination. This phase involves teams configuring Snort so that it will inspect the packet and if it matches any of the rules, Snort will drop the packet instead of allowing it to continue on.

In configuring Snort as a NIPS, students learn how to move the DMZ traffic through the Snort VM. This requires creating a layer 2 bridge for network traffic to cross. Teams create a new virtual switch on the ESX server to place the Snort NIPS VM between the firewall and DMZ; see Figure 2. To verify a correct implantation of the NIPS, students try to access outbound or inbound web traffic with a word that has been denied in phase 6. If the requests are dropped then teams were successful. This phase is accomplished during week 10.

ISBN 1-933510-96-X/$15.00

IX. Phase 8

Previously, phase 8 was having students harden their infrastructure and then perform an IT audit on different team's systems. Recently this phase has changed to set up a SNARE server to perform log analysis from all of the different security systems created throughout the course. The teams SNARE VM will receive logs from firewalls, web servers, Freeradius, OSSEC, Snort, and Squid.

One of the problems facing security professionals is bringing together data from many different systems to provide meaningful information. This phase will help students understand the complexity of maintaining and managing disparate logs in order to effectively monitor cybersecurity systems. By collecting all of the security data into one location with a tool that can provide meaningful results, students will learn how to look at the entire environment as a whole and not just separate parts. Students will spend week 11 and 12 working on this phase.

X. Final Project

After successfully completing all eight phases of the course a final project will include a penetration test. Students will have spent 12 weeks learning how to defend against intruders by installing and configuring systems to protect their organizations. During the final project, students will gain experience using tools to penetrate systems. This experience will help students understand what they are up against and where in the "seams" attacks occur.

Each team will be given a vulnerable VM to attack. Behind each VM are 3 secured VMs, see Figure 3. The only information teams are provided is to find 3 pieces of personal identifiable information (PII). Each team is provided an IP address of their vulnerable system. From that point on, teams try to scan and penetrate their vulnerable system and then pivot to find the PII.

It is fascinating to watch students dive into this project. They will forgo sleep and even other classes in order to be the first ones to find the PII. In order to successfully complete the final project students will need to exploit their team's computer and download the user accounts and passwords. After cracking the passwords for these accounts, they will then use these accounts to access the three secured servers, which contain the PII.

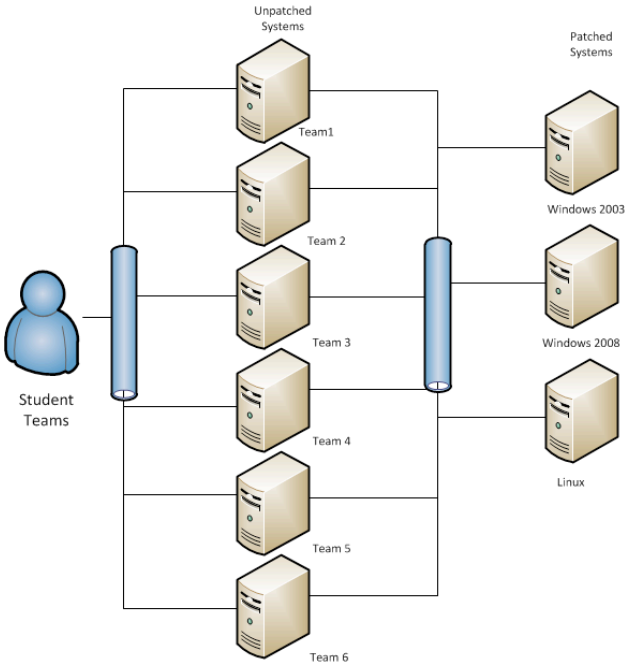

Figure 3.

XI. Assessment

Each phase of the course is assessed using three criteria. The first criterion is whether the teams successfully accomplished all of the stated requirements for the phase. Each team score's the final results of the phase with the instructor verifying this score by examining the team's virtual environment. The second criterion is a group evaluation of each team member's performance during the phase. Students will score each member of the team by distributing points. For example, a team of 3 students will be given 20 points to divide among their members. The last criterion is the number of hours they committed to the phase. Each student is required to keep a journal documenting the activities and time spent on each phase.

XII. Summary

This paper begins by discussing the increasing need for cybersecurity professionals with the ability to understand the interconnecting parts of security systems. A new lab-based course design is described that provides students an opportunity to see how complex cybersecurity systems work together. This course design includes eight phases with one final project. Each phase provides students the opportunity to see how security systems work together and rely on the other in order to obtain the "big picture" of the overall security of an organization. A final project at the end of the course gives students opportunities to see how systems are attacked by doing a pen test on a Virtual network with unsecure and secure systems. Through these 8 phases and final project, students are able to build

ISBN 1-933510-96-X/$15.00

a cybersecurity system in its entirety with an understanding how these systems work together.

XIII. References

[1] National Coordination Office for Information Technology Research and Development, "President's Information Technology Advisory Committee," www.nitrd.gov/pitac (26 May 2005).

[2] Brown, S., & Lahoud, H. (2005). An Examination of Online Lab Technologies. Proceedings of the ACM SIGITE 2005 Conference, Newark, NJ.

[3]Leitner, L., & Cane, J. (2005). A Virtual Laboratory Environment for Online IT Education. Proceedings of the ACM SIGITE 2005 Conference, Newark, NJ.

[4] Padman, V., Memon, N., Frankl, P., and Naumovich, G. (2003). Design and Implementation of an Information Security Laboratory. Proceedings of World Conference on Information Security Education.

[5]Stockman, M. (2003). Creating Remotely Accessible "Virtual Networks" on a Single PC to Teach Computer Networking and Operating Systems. Proceedings of the 2003 CITC, Lafayette, IN.

[6]Toderick, L., Mohammed, T., Tabrizi, M. (2005). A Consortium of Secure Remote Access Labs for Information Technology Education, Proceedings of the ACM SIGITE 2005 Conference, Newark, NJ.

[7]Yang, T., Yue, K., Liaw, M., Collins, G., Venkatraman, J., Achar, S., Sadasivam, K., Chen, P. (2004). Design of a distributed computer security lab. Journal of Computing Sciences in Colleges, 20 *(1)*, pp. 332-347.

ISBN 1-933510-96-X/$15.00