Teaching Software Security to Novices With User Friendly Armitage

Christopher Morales-Gonzalez
Miner School of Computer & Information Sciences
University of Massachusetts Lowell
Lowell, MA, USA
Christopher_MoralesGonzalez@student.uml.edu
0000-0001-9403-6837

Matthew Harper
Miner School of Computer & Information Sciences
University of Massachusetts Lowell
Lowell, MA, USA
Matthew_Harper@student.uml.edu
0000-0003-1749-7976

Xinwen Fu
Miner School of Computer & Information Sciences
University of Massachusetts Lowell
Lowell, MA, USA
xinwen_fu@uml.edu
0000-0003-2391-7789

***Abstract*—With cybercrime increasing by 600% during the COVID-19 pandemic, the demand for cybersecurity professionals has also risen significantly. There are roughly 700,000 unfilled cybersecurity positions that continue to affect businesses and have the potential to cause significant problems. Education for novice cybersecurity students suffers from teaching materials not being practical, modern, nor intuitive enough to inspire these students to pursue a career in the cybersecurity field. In this paper, we present our methodology and create a module for teaching the basics of software security using Armitage and Metasploit. We design our module and hands-on labs using a preconfigured Windows 10 VM, a Metasploitable VM and a Kali Linux VM with custom-made tools. Our methodology and module is validated through the results of a GenCyber high school cybersecurity camp. The module is available at GitHub[1].**

Keywords—Software Security, Metasploit, Armitage

I. INTRODUCTION

Cybercrime is becoming increasingly common as the world is becoming ever more interconnected. There are various forms of cybercrime such as phishing, fraud, cyberextortion and data exfiltration. It is reported that because of the COVID-19 pandemic, cybercrime activities have increased by 600% [1]. Recent cybercrime attacks such as the ransomware attack on Costa Rica [2] or the data breach Saudi Aramco suffered [3] highlight the reality of the threat of cybercrime.

To counter, defend and respond to cybercrimes, there must be enough cybersecurity professionals. However, there is a major shortage of these professionals in the US. It is reported that there are over 700,000 cybersecurity job openings in the US in 2022 [4]. This shortage emphasizes the importance of inspiring and motivating students to pursue higher education and a career in cybersecurity.

There is a lack of high-quality teaching materials for novice cybersecurity students to motivate them to pursue advanced cybersecurity education and careers. Cybersecurity has a broad range of topics such as cryptography, network security, digital forensics and software security. Software security is a hard topic given that C and assembly language are often needed to master advanced software security concepts such as the buffer overflow attack. The C and assembly languages are challenging for novices. However, we still want to demonstrate software security to novices with proper depth given that vulnerabilities in network-facing software can lead to problematic exploits that can have serious consequences if left unchecked and unsecured.

In this paper, we focus on teaching novices basic concepts of software security so that they can see the potential impacts of attacks and be inspired. We have developed a custom module using Armitage and Metasploit to facilitate an easy learning environment. The module is created with the intent to be practical, realistic, modern, accurate and intuitive. Armitage provides an intuitive graphical user interface to use the advanced security and hacking tool—Metasploit. The hands-on labs based on Armitage are both intuitive and meaningful, and directly reflect the theoretical concepts and showcase the practical impacts of software security. These attributes are what Williams and Williams describe to be crucial to create an environment that will motivate students to learn [5]. The end goal is to inspire students to pursue a career within the field of cybersecurity and equip them with a valuable skill set to aid in the ongoing battle against cyber criminals.

Our module begins with teaching students about the cyber attack cycle to gain an understanding of the three phases of a cyber attack: information gathering, exploitation and post-exploitation. Teaching the big picture of a cyber attack with proper examples is critical to the understanding of various attacks and defenses since students are often confused by how different knowledge units scattered across different courses or tutorials can be put together. After the cyber attack cycle is discussed, then an introduction to the penetration testing tools Metasploit and Armitage is given. These tools are specifically chosen to expose the students to real-world tools used in the professional cybersecurity world. Finally, a live demonstration of the cyber attack cycle is performed using these tools to showcase the feasibility of a cyber attack.

This work was partially supported by an NSA GenCyber grant. Any opinions, findings, conclusions, and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

[1] https://github.com/xinwenfu/GenCyber/tree/main/SoftwareSecurity

We create our own vulnerable chat server [6], and custom Metasploit modules [7] to use during the demonstration and hands-on labs. These Metasploit modules are developed in Ruby, can run under Armitage within Kali Linux, and perform a buffer overflow attack, Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attack against our vulnerable chat server, which runs on latest versions of Windows 10 or 11. Having developed these tools to work on modern Linux and Windows systems will keep the content of the module modern, realistic, and accurate. The Metasploit modules also make the lab more intuitive because it will remove the coding requirement for performing these attacks and will maintain the focus on the premise of the attack while we provide example Ruby and Python code for curious students to try programming the attacks themselves.

We create six hands-on labs for the software security module and evaluate the effectiveness of this module at a GenCyber high school cybersecurity camp in Summer 2022 hosted at our university. Students finished the hands-on labs and a capture-the-flag (CTF) competition at the end of the camp successfully.

Our major contributions can be summarized as follows:

- We present a simple, linear way of teaching students about the basics of software security in an intuitive manner. This method starts with introducing the cyber attack cycle to understand how an attacker thinks, then it continues onto an introduction to the popular penetration testing tools Armitage and Metasploit with our custom modules, and lastly a demonstration of the cyber attack cycle using these tools is given.
- We create our own novel vulnerable chat server, which students can use to chat with each other. Student can use our custom Metasploit modules that perform a buffer overflow for hacks such as camera streaming, DoS and a DDoS attack against the vulnerable chat server.
- Our methodology and module were validated during a GenCyber high school cybersecurity camp hosted at our university during Summer 2022. The hands-on labs and CTF require knowledge of the cyber attack cycle as well as Armitage and Metasploit. Students performed well at the camp and CTF.

II. SOFTWARE SECURITY MODULE

The module presented in this paper is broken into four components: i) cyber attack cycle, ii) introduction to Metasploit and Armitage, iii) demonstration, and iv) hands-on labs.

A. Cyber Attack Cycle

To be able to mitigate and defend against attacks, cybersecurity professionals must be able to think like an attacker. Understanding the phases that an attacker will go through is crucial to developing appropriate and effective defenses. Cyberattacks typically follow the three major phases shown in Fig. 1: i) Information collecting, ii) Exploitation, and iii) Post exploitation. Please note Fig. 1 shows some examples of attacks in each phase and does not try to be comprehensive.

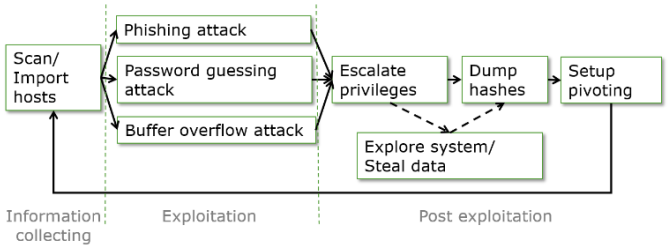

Fig. 1. Cyber Attack Cycle

i) Information Collecting: To carry out an attack, an attacker must gather information regarding their target. This collection can be carried out in many ways; an example is through port scanning. Port scanning will allow the attacker to find open ports on a target machine and may be able to find out about the different services running on the target. The attacker has the option of scanning a range of addresses to gather information on them and determine a larger number of possible targets.

ii) Exploitation: With the targets and information gathered about them, the attacker will now enter this phase to start exploits. The attacker can deploy a remote attack to one of these hosts to try and gain access to the target machine. This can be in the form of a brute-force attack where the attacker tries to continuously guess the credentials of a user on the machine to gain access. The buffer overflow attack is another example of remote attack and can be used to inject malicious code into a target server. Another possibility is that the attacker will try to perform a client-side attack in which the attacker lures a user of the victim machine to download a piece of malware that will execute upon some action. For example, this could be done in a social engineering attack that would send an email that has an attachment containing a malicious piece of code embedded within it. All is done in the hopes that some form of access is gained within the target.

iii) Post-Exploitation: Once the attacker gains access to one of the target machines, there can be a myriad of ways that an attacker can cause more damage. A common path is through privilege escalation. Within this, the attacker may try techniques to gain root access to have full control of the system. With this root access gained, the attacker has full access to the file system. This has the potential to expose sensitive files that can be potentially exfiltrated and incur financial costs for the affected target. One of these sensitive files can be the password hashes file. Once this file is taken, the attacker can then crack the passwords offline using a tool like John the Ripper.

Another method that can be performed is setting up persistent access to the machine. Once the attacker gets into the machine, they can run malware directly on that machine since they have the same access a user would. For example, they could setup a *backdoor* program that would allow the attacker to access the machine even upon a restart. This

device can be used as a foothold into the internal network if this machine is part of an enterprise network. Once there, the attacker can perform the entire cycle again, but against internal machines that wouldn't originally be publicly accessible.

B. Introduction to Metasploit and Armitage

With the mentality of an attacker in mind, we now introduce Armitage and Metasploit, which is very popular amongst the cybersecurity community.

1) Metasploit: The Metasploit Framework [9] is used by cybersecurity professionals to perform penetration testing. It is made up of numerous different modules that can perform entire exploits or specific portions of one exploit. It will eliminate the requirement of having a novice understand the intricacies of an attack and how to perform or program the attack. The early stages of this framework could only carry out actions by invoking command line statements. If we are trying to teach novice cybersecurity students, a command-line-only interface would prove to be detrimental to their learning as it is not intuitive and is error-prone due to syntactical errors.

2) Armitage: Armitage solves the unfriendly user interface problem of Metasploit by providing an intuitive graphical user interface (GUI) as shown in Fig. 2 for Metasploit. Whatever the user chooses from the Armitage GUI, Armitage will map those commands into the equivalent Metasploit commands and run them automatically. This eliminates the need for the student to remember syntax, flags and other arguments that may be needed when running an attack from the Metasploit framework.

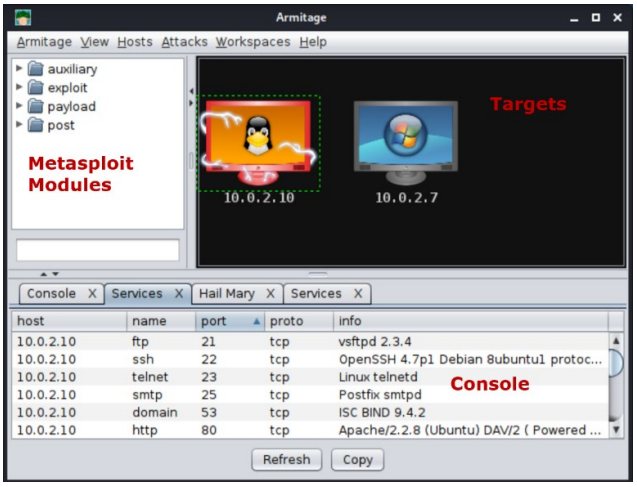

Fig. 2. Armitage, GUI front end of Metasploit

To harness the full power of Armitage, Armitage must be run with elevated privileges. Armitage is capable of performing various types of port scans that can be used by an attacker for information gathering. With the information gained from the port scanning, Armitage provides a feature called "Find Attacks" that will consult the Metasploit Exploit Database to match the service version with a known exploit. If there is a match, then Armitage will give the user the option on the GUI to perform the specific attack on the target. Having this feature will only increase the efficiency of showing the final result, which can further inspire students to learn more.

C. Demonstration of Cyber Attack Cycle

1) Setup

Now we transition into giving a demonstration of the full cyber attack cycle using Armitage and Metasploit. First, we describe the details of the virtual machines (VMs) used. Then we describe our modern vulnerable chat server. Finally, we provide insight into the custom Metasploit modules that are created for this software security module.

i) Metasploitable 2: We use the Metasploitable 2 [10] VM provided by Rapid7 as one of our victim machines. This VM is specifically designed to have a multitude of vulnerabilities present to be used for practice. This serves as a way for students to practice responsibly and safely.

ii) Kali Linux: We use the Kali Linux VM provided by Offensive Security [11] as the attacker machine. This VM comes with Metasploit pre-installed as part of the image. We must manually install Armitage onto the system as it does not come preinstalled. Kali Linux is a popular cybersecurity learning platform given its various pre-installed tools.

iii) Windows 10: We preconfigure a Windows 10 VM with its entire *exploit protection* settings, *Windows Defender Firewall* settings, *Virus & Threat Protection* settings, and *Real-Time Protection* security functionality turned off. Having this done will prevent Windows from detecting and neutralizing malicious code that is being executed. This VM will host our vulnerable chat server.

These three virtual machines above are hosted within Oracle's VirtualBox [12] hypervisor and are configured to be able to communicate with each other using the "NatNetwork" network type.

iv) Vulnerable chat server: We created our own novel vulnerable chat server in C that works on Windows 10 and 11 and contains specific vulnerabilities intentionally left for penetration testing purposes. We use this tool to showcase that seemingly working services such as a chat server on the latest Windows can have potentially severe vulnerabilities that can be exploited. Having this new software keeps the module modern and relevant while providing the capability to extend the tool to encompass any other vulnerability that we may want to teach in the future.

v) Metasploit modules: Metasploit is made up of different modules from the open-source community. Metasploit provides users with a way of creating their own modules. This allows us to showcase specific attacks, and provides us a way to show students that what they're learning can be directly applied into a real-world scenario.

Specifically, we create three Metasploit modules that work with Armitage. i) Buffer overflow module: The vulnerable chat server uses a buffer that can only handle messages of 2000 characters or less. Buffer overflow occurs

when an attacker sends a long message that exceeds the size of the buffer. This message may contain malicious code in its attack payload. The module uses Metasploit's Ruby classes and can incorporate various Metasploit attack payloads. ii) DoS module: The buffer overflow vulnerability can be used for a DoS attack. The DoS module will send a message that is 10,000 characters long and will cause the server to crash. The crashing of the server will result in denial-of-service as no more legitimate connections can be established with the chat server. iii) DDoS module: The vulnerable chat server is intentionally designed to only handle 100 concurrent client connections. With this, we can see a potential DDoS vulnerability if the server does not have the ability to scale or remove inactive connections. Multiple students may use the DDoS module to attempt to establish more than 100 sessions with the server. Once this is done, then no new legitimate connections can be established to the server, thus a denial-of-service is performed.

2) Demonstration

Understanding the theoretical portion of software security is important. However, a crucial piece is an intuitive hands-on demonstration that reinforces what the students just learned. This will show that the content they learned can be directly translated into a real-world situation and students can see the consequences of poor software security. This will showcase the power of Armitage against the vulnerable Metasploitable machine and our vulnerable chat server.

During our demonstration, we seek to perform every step of the cyber attack cycle. For the information gathering phase, Armitage on the Kali Linux attacker machine will perform an msf or nmap scan against the Metasploitable victim machine. Upon doing this, the attacker will learn the services running on the machine. We then show the convenience of the "Find Attacks" function that attempts to find known exploits against these particular services running on the Metasploitable machine.

Now we transition to demonstrating the exploitation phase on the Metasploitable machine first. The attacker now chooses the *unreal_ircd_3281_backdoor* exploit that affects the Unreal IRCD 3.2.8.1 server [13], which has a backdoor planted. Upon successful execution of the attack, the GUI indicates that the attack was successful by changing the icon of the machine.

With a successful attack and interactive shell gained, we now show what possible post-exploitation steps can be taken by an attacker. Particularly, we choose the "dump the hash" path. The attacker will dump the password hashes of the Linux-based machine into the console and then they will simply copy-and-paste them into a text file. All the attacker needs to do is within the "post" folder under Armitage's module panel, they simply need to select the "hashdump" module which will place the hashes onto the screen. Now that they have extracted the hashes, the attacker can use John the Ripper [14] to crack them and obtain user credentials on the machine.

After showing the attack against the Metasploitable VM, we can also show the attack against the vulnerable chat server on the Windows 10 VM. During the information collecting stage, an msf/nmap scan will indicate that the target port 9999 for the vulnerable chat server is open. For the sake of simplicity, we assume that no other programs that utilize this port are running. Now, we deploy our Metasploit modules discussed before against the Windows machine. First, we demonstrate the DoS attack. Armitage's GUI allows us to simply double-click the module, enter the IP address of the Windows machine and then press run. On the Windows machine, the vulnerable chat server will crash indicating that the attack was successful. Upon a restart of the vulnerable chat server executable, then we deploy the DDoS attack. The process is the same as DoS but with an extra parameter to indicate how many sessions the user would like. We provide this flexibility for students to experiment and see different behaviors given different inputs. Once we run the attack, when a new legitimate connection is trying to be made, then the vulnerable chat server will indicate that it cannot handle more connections. The buffer overflow attack can also be easily demonstrated. A few clicks will allow us to hack into the Windows 10 VM and perform post exploitation tricks such as camera streaming, keystroke logging and screenshot capturing.

D. Hands-on Labs

The following hands-on labs are created to assess the learning of the developed software security module:

- *Hands-on 1*. Hacking into the Metasploitable 2 VM through *unreal_ircd_3281_backdoor*.
- *Hands-on 2*. Hacking into the Windows VM through a buffer overflow attack on the vulnerable chat server using a pre-created buffer overflow Metasploit module.
- *Hands-on 3*. Capturing a screenshot through Metasploit meterpreter payload.
- *Hands-on 4*. Keystroke logging through Metasploit meterpreter payload.
- *Hands-on 5*. Deploying persistent backdoor.
- *Hands-on 6*. Buffer overflow attack via Python code. This lab is designed for students who are familiar with Python and want to delve further into the buffer overflow/DoS/DDoS attack. The Python code is very similar to the Ruby code in the corresponding attack. However, the Ruby module for Metasploit allows the selection of various Metasploit attack payloads.

III. EVALUATION

In this section, we first introduce the class environment and then show the results of evaluating the effectiveness of the module presented in Section II on teaching software security.

A. *Class Setup*

We ran a one-week GenCyber cybersecurity camp for 40 high school students from grades 9 – 12 in 2022 summer. The developed software security module was used at the camp and the students did the six hands-on labs after the lecture on the module.

At the end of the camp, we ran a Capture-the-Flag (CTF) competition that contained a software security challenge that covered this module. During the CTF, students were encouraged to form teams of up to two individuals to work together and complete the CTF to promote collaboration and develop crucial communication skills. Each team was asked to submit their answers via a survey that was given through Google Forms. The results highlight areas in which the students are both fluent and not fluent in. This indicates the effectiveness of the module.

B. *Learning Effectiveness*

With the help of teaching assistants, all camp students finished the developed Hands-on Labs 1-5 and some students were also able to finish the more advanced Hands-on Lab 6.

34 students showed up on the day of CTF. Upon closing the CTF for submissions, the results were calculated, and the software security question (hacking Metasploitable 2, dumping and cracking password hashes) was the one of the most successful flags in terms of correctness out of the entire competition. We count how many students did the software security question correctly, how many did not respond (DNR) (i.e., did not answer the software security question), and how many did not participate (DNP) in the CTF (i.e., did not submit any answer at all). The software security question had received an 82.4% correctness rating, an 8.8% DNR rating and an 8.8% DNP rating as shown in Fig. 3. This proves the effectiveness of the module because most students were able to understand the material in a short amount of time and could recreate the hands-on lab that they were assigned earlier.

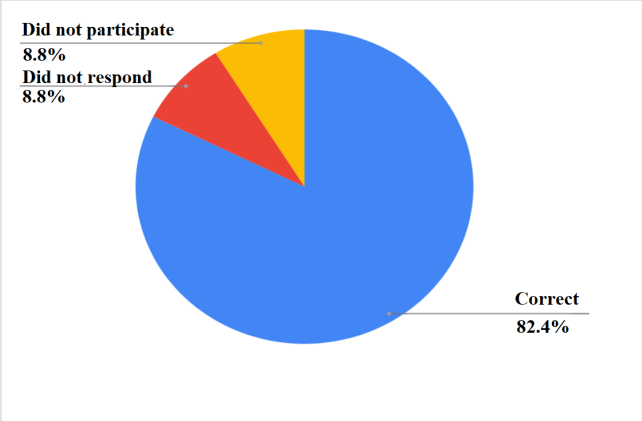

Fig. 3. CTF Software Security Question on hacking Metasploitable 2, dumping and cracking password hashes

IV. CONCLUSION

In this paper, we present our methodology and module for teaching the basics of software security to novice cybersecurity students. We begin by explaining all phases of the cyber attack cycle: information collecting, exploitation and post-exploitation. We then introduce the popular penetration testing tools Metasploit and Armitage with our own custom attack modules and use these tools to provide a demonstration of how a cyber crime can be carried out. We evaluate the effectiveness of this methodology and module through the assessment of hands-on labs and a CTF competition specifically designed to reflect the learning of the module during a GenCyber high school cybersecurity camp. We show that this module is effective in teaching the basics of software security as it was one of the highest rated flags in term of correct answers.

ACKNOWLEDGEMENT

This research was supported in part by US NSA GenCyber grants 21A-MA-UMLx-UV-S1 and 21-MA-UMLx-UV-S1, and US National Science Foundation (NSF) Awards 1931871 and 1915780. Any opinions, findings, conclusions, and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

REFERENCES

[1] "2022 cyber security statistics trends and data," *PurpleSec*, 18-Jul-2022. [Online]. Available: https://purplesec.us/resources/cyber-security-statistics/#Cybercrime. [Accessed: 09-Sep-2022].

[2] B. Krebs, "Costa Rica may be pawn in Conti ransomware group's bid to rebrand, evade sanctions," *Krebs on Security*, 31-May-2022. [Online]. Available: https://krebsonsecurity.com/2022/05/costa-rica-may-be-pawn-in-conti-ransomware-groups-bid-to-rebrand-evade-sanctions/. [Accessed: 09-Sep-2022]

[3] J. Allen, "Saudi Aramco $50 million data breach explained," *PurpleSec*, 06-Aug-2021. [Online]. Available: https://purplesec.us/saudi-aramco-data-breach-explained/. [Accessed: 09-Sep-2022].

[4] Cybersecurity Supply/Demand Heat Map. [Online]. Available: https://www.cyberseek.org/heatmap.html. [Accessed: 09-Sep-2022]

[5] K. C. Williams and C. C. Williams , "Five key Ingredients for Improving Student Motivation ," *Research in Higher Education Journal*, Aug-2011. [Online]. Available: https://aabri.com/manuscripts/11834.pdf. [Accessed: 09-Sep-2022].

[6] Xinwen Fu, "Vulnerable Chat Server (vchat)," Malware Analysis vchat, 31-May-2022. [Online]. Available: https://github.com/xinwenfu/Malware-Analysis/tree/main/vchat. [Accessed: 09-Sep-2022].

[7] Matthew Harper and Xinwen Fu, Making DoS and DDoS Metasploit modules. [Online]. Available: https://github.com/DaintyJet/Making-Dos-DDoS-Metasploit-Module-Vulnserver. [Accessed: 09-Sep-2022].

[8] Nmap. [Online]. Available: https://nmap.org/. [Accessed: 09-Sep-2022].

[9] "Metasploit framework," *Metasploit Framework | Metasploit Documentation*. [Online]. Available: https://docs.rapid7.com/metasploit/msf-overview/. [Accessed: 09-Sep-2022].

[10] "Metasploitable 2," *Metasploitable 2 | Metasploit Documentation*. [Online]. Available: https://docs.rapid7.com/metasploit/metasploitable-2/. [Accessed: 09-Sep-2022].

[11] "Penetration testing and ethical hacking linux distribution," *Kali Linux*. [Online]. Available: https://www.kali.org/. [Accessed: 09-Sep-2022].

[12] "Virtualbox" *Oracle VM VirtualBox*. [Online]. Available: https://www.virtualbox.org/. [Accessed: 09-Sep-2022].

[13] HDM, "UNREALIRCD 3.2.8.1 backdoor command execution - metasploit," *InfosecMatter*. [Online]. Available: https://www.infosecmatter.com/metasploit-module-library/?mm=exploit%2Funix%2Firc%2Funreal_ircd_3281_backdoor. [Accessed: 09-Sep-2022].

[14] "John the ripper password cracker," *Openwall*. [Online]. Available: https://www.openwall.com/john/. [Accessed: 09-Sep-2022]