Jeffery Dirden
ITAI 1372
October 16, 2024

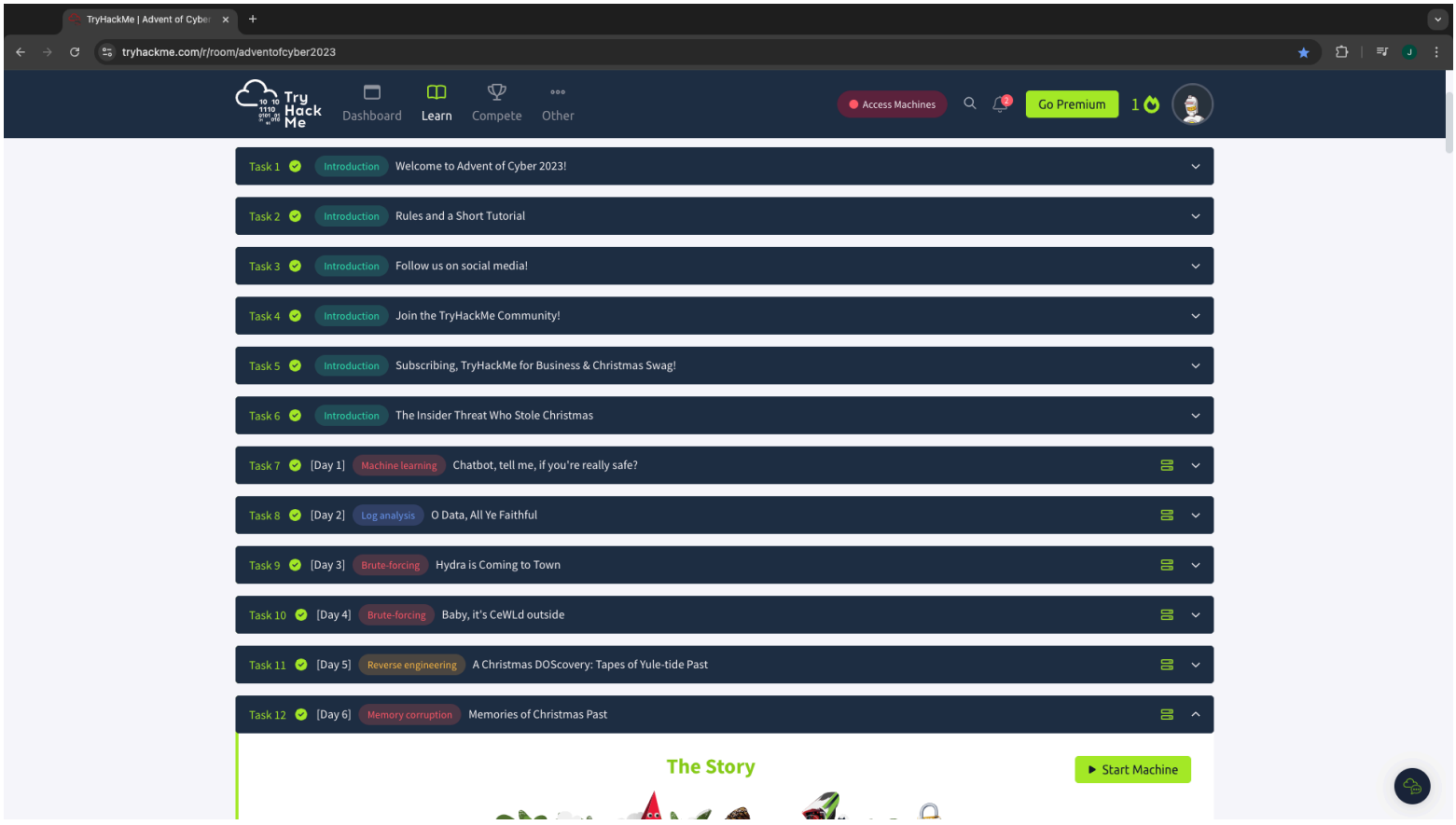
Midterm

Using the Tryhackme platform has been very beneficial for me, gaining a more in depth understanding of how cybersecurity works, getting my feet wet and gaining experience launching the virtual machines has been fun as well. I really appreciate the fact that labs are hands on with good detailed readings, videos, and visual representations it really has made things more interesting and interactive for this class, and makes it a easier for me to retain the information being discussed. Now I'll discuss what I learned from the 3 labs that I've decided to talk about.

Day 3 is a lab called "Hydra is coming to town. It focuses on brute-forcing which is. It put in perspective the complexity and possible combinations for passwords. How the number of potential combinations affects the feasibility of brute force attacks. Generating password combinations using crunch, and also trying passwords automatically using hydra. I learned a lot about the basics of brute force attacks in the penetration testing lab, including the process of carefully guessing credentials to obtain access to systems. Task 9 showed how brute-forcing operates by using tools such as Hydra to try username and password combinations. I also learnt how to spot weak places in entrance and reduce the danger by implementing security measures like CAPTCHA systems and account lockouts. Using these abilities, I can test password regulations in my company, reducing the possibility of credential based attacks and ensuring strong password security. Putting in place the right security measures, such multi-factor authentication, would be essential to preventing brute force attacks.

Day 5 focuses on reverse engineering in a lab called "A Christmas DOScovery: Tapes of Yule-tide past". The objectives of this lab was to gain experience and get familiar to learn how to navigate unfamiliar legacy systems. Gain insight on DOS and its connection to the windows

command prompt. Also discovering the importance of file signatures and magic bytes in data recovery and file system analysis. I learned about digital forensics in this lab, which includes gathering and examining information from infected systems. I gained insight into how to analyze disk images, retrieve erased data, and use system logs to monitor criminal behavior. In order to determine how a breach happened, what data was impacted, and how to prevent similar assaults in the future, the lab highlighted the vital role that forensics plays in incident response. By examining a program to understand its operation without having access to its source code, Task 11 introduced reverse engineering. I learnt how to analyze malware activity, retrieve secret information, and decode binary files. By dissecting executable files and analyzing their internal principles to determine how attackers take advantage of them, this lab gave me a serious perspective into spotting software vulnerabilities. Proficiency in reverse engineering is vital for malware analysis and software security evaluations. I would use my skills in a work setting to evaluate potentially harmful software, comprehend its workings, and create plans to fix



flaws. By understanding the issue and creating tools to detect and stop more attacks, this could help in reacting to assaults.

Day 6 focus on memory corruption in a lab called “Memories of Christmas Past”. Here I learned how certain languages handle memory some better than others. How variables can overflow in adjacent memory and corrupt it, and using basic buffer overflow to change memory you are not suppose to access. This lab focused on digital forensics, I had to retrieve and examine data from infected computers. I learned how to investigate log files, understand file system structures, and create and analyze disk images. This is important for determining the origin and extent of an attack and for protecting evidence for compliance and legal reasons. I could use these abilities to audit software for buffer overflow vulnerabilities in a business environment, making sure that the apps used by my company are safe from such assaults. In order to reduce the dangers of memory corruption, I would favor secure coding techniques including input validation and the usage of memory safe programming languages.