Python for Cybersecurity

Assignment journal

Student registration number	Student name
It21822612	Mendis H.R.M.
tryhackme.com/jr/pythonforcybersecuritysnp	

IE2012- system and network programming

Date of submission: 5/24/2023

Introduction

Welcome to my assignment journal, where we will look at real Python applications in the topic of Cyber Security. This journal will cover three major topics: network scanning with Python-Nmap, web scraping with Scrapy, and exploit creation with Pwntools. Understanding Python's strengths in these areas allows us to improve our cyber security policies and create helpful solutions to defend digital environments.

Our journey begins with a network scan using Python-Nmap. We can efficiently detect open ports and gain essential data about target networks by automating network scans. Python-Nmap gives us the tools we need to do effective network reconnaissance and boost our security measures.

After that, I'll look at site scraping using Scrapy, a strong Python framework. We can extract data from websites using Python's capabilities, which is critical for cyber security investigation. Scrapy allows us to traverse online sites, collect pertinent information, and evaluate the data returned to detect potential hazards.

Finally, we'll look at exploit development with Pwntools. Exploit development is critical for understanding and protecting against software vulnerabilities. Pwntools, in conjunction with Python, provides us with the ability to find vulnerabilities, build exploits, and produce payloads, allowing us to safeguard our systems proactively.

We will blend theoretical knowledge with practical exercises throughout this tryhackme session to increase our grasp of Python's role in cyber security. By the end of this course, you will have learned useful insights and practical skills for improving your cyber security procedures with Python-Nmap, Scrapy, and Pwntools. Let us begin this instructive journey to unleash Python's potential for cyber security excellence!

Why did I choose this topic?

I chose the project because I had some knowledge of Python from past academic studies, specifically my A/Levels. Python's simplicity and flexibility have always caught my interest, and I believe that improving my understanding of Python libraries, particularly in the context of cyber security, will be both beneficial and enjoyable.

With my basic knowledge of Python, I am excited to learn about its application in the field of cyber security. This topic allows me to build on my previous knowledge while delving deeper into certain Python libraries that are commonly used in the sector.

Furthermore, I believe that the study will provide me with practical tools and strategies that are in great demand in the ever-changing world of cyber security. The ability to use Python to automate network scans, gather data from websites, and construct exploits can significantly improve the security posture of digital environments.

Date: may 10 2023

Today's Goals:

Read the assignment and find Python libraries that are appropriate for this task.

Begin by researching the following libraries: Python Nmap, Scrapy, and pwntools.

Goals Accomplished:

Python Nmap, Scrapy, and pwntools were identified as suitable libraries.

I began researching libraries.

Challenges:

The concepts and functionalities were difficult to understand due to a lack of prior knowledge of the specified libraries.

Balancing many assignments at the same time added to the difficulty of focusing on these libraries.

I started working on the assignments today, which required me to look into Python libraries for network scanning, web scraping, and exploit development. My first goal was to carefully examine the assignment instructions and identify which Python libraries were appropriate for each task. I identified Python Nmap for network scanning, Scrapy for web scraping, and pwntools for exploit building after analyzing the requirements.

After identifying the libraries, I began researching online resources. I rapidly realized, however, that I had little to no prior understanding of these libraries, which provided a hurdle. It took extra effort and patience to learn about their features, syntax, and implementation.

Another challenge I faced was balancing my time and attention between many assignments. While I aimed to concentrate on researching the libraries identified, I also had other obligations and deadlines to complete. Finding the correct balance between them has become critical in order to make progress on all fronts.

Whatever of the difficulties, I was determined to overcome them. I set aside time to examine each library, looking for tutorials, guides, and examples that would assist me obtain a better understanding. I took notes and experimented with code snippets to ensure my understanding.

Resources and References:

https://pypi.org/project/python-nmap/

https://docs.scrapy.org/en/latest/intro/tutorial.html

Date: may 11 2023

Today's Goals:

Today, I wanted to focus on Python-Nmap and do research to better understand this library. I chose to investigate Python-Nmap because I was already familiar with Nmap and thought it would be a simple step to using Python for network scanning.

Goals Accomplished:

During my study, I learned about Python-Nmap and its features. I learned how to run Python-Nmap as a foundation for conducting network scans. I was able to add Nmap switches and tweak the scanning procedure using this example.

Challenges:

One of the difficulties I experienced was limited availability of Python-Nmap-specific learning resources. I was able to solve this challenge by using Chat GPT as a source of knowledge and guidance. Chat GPT gave me good insights and helped me understand Python-Nmap better.

Overall, today's Python-Nmap research was a pleasant experience. I accomplished my goal of learning about the library and its basic functions. Despite the difficulties in locating learning tools, I was able to improve my understanding by using alternate approaches like as Chat GPT. I'm excited to continue my adventure and learn more about Python for cyber security.

Date: may 12 2023

Today's Goals:

My goal for today was to continue my research on Python-Nmap and look at its practical uses in network scanning for cyber security.

Goals Accomplished:

To obtain hands-on experience, I attempted to install Python-Nmap on my machine. I was curious to explore how the library could automate network scans and deliver useful information about network vulnerabilities.

Challenges:

However, I faced some difficulties along the process. I faced installation problems with Python-Nmap, primarily due to compatibility issues with the Python version I had installed on my machine. It proved to be an issue in my initial attempts to get Python-Nmap up and running.

Moving forward, I intend to fix the installation issues that I faced. I'll investigate several Python versions and troubleshoot Python-Nmap compatibility issues. In the coming days, my goal is to overcome these obstacles and continue my investigation of network scanning with Python-Nmap.

Resources and References:

https://www.geeksforgeeks.org/how-to-install-python-nmap-library-in-linux/

https://pypi.org/project/python-nmap/

Date: may 13 2023

Today's Goals:

Learn how TryHackMe rooms work.

Understand the process of creating a room in TryHackMe.

Upload a virtual machine (VM) to TryHackMe.

Goals Accomplished:

Today, I achieved my goal of creating my first TryHackMe room.

Challenges:

Understanding how uploaded VMs functioned in TryHackMe was one of the most difficult issues I faced. I wasn't sure how to start and stop the tasks linked with the uploaded VM at first. However, on the TryHackMe website, I came upon a great blog post titled "Making the Mountain." This site provided detailed guidance for building and managing rooms, which assisted me in overcoming this difficulty.

The evaluation of TryHackMe today was a fantastic learning experience. I obtained a better understanding of how TryHackMe rooms work, which enable users to participate in practical cyber security problems. Creating my first room was a huge accomplishment since it allowed me to construct and structure my own tasks for others to solve.

While there were some difficulties, such as comprehending the complexities of uploading VMs, the materials given on TryHackMe, such as the instructive blog post, proved essential in overcoming these problems.

Resources and References:

https://tryhackme.com/resources/blog/making-the-mountain

Date: may 14 2023

Today's Goals:

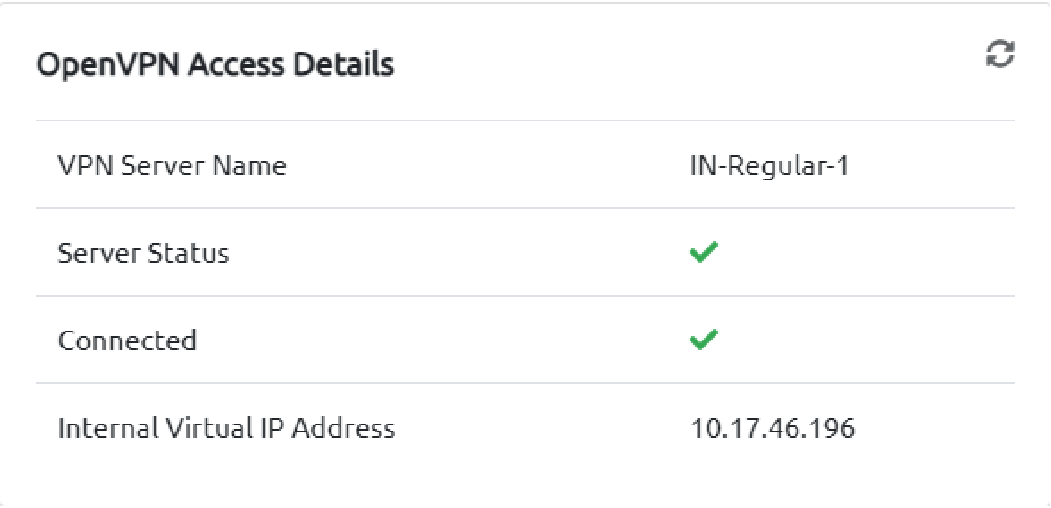
Connect my own machine to the tryhackme network.

Create the first task explaining how to connect your PC to the network.

Develop a task to teach the basics of using the Nmap tool.

Goals Accomplished:

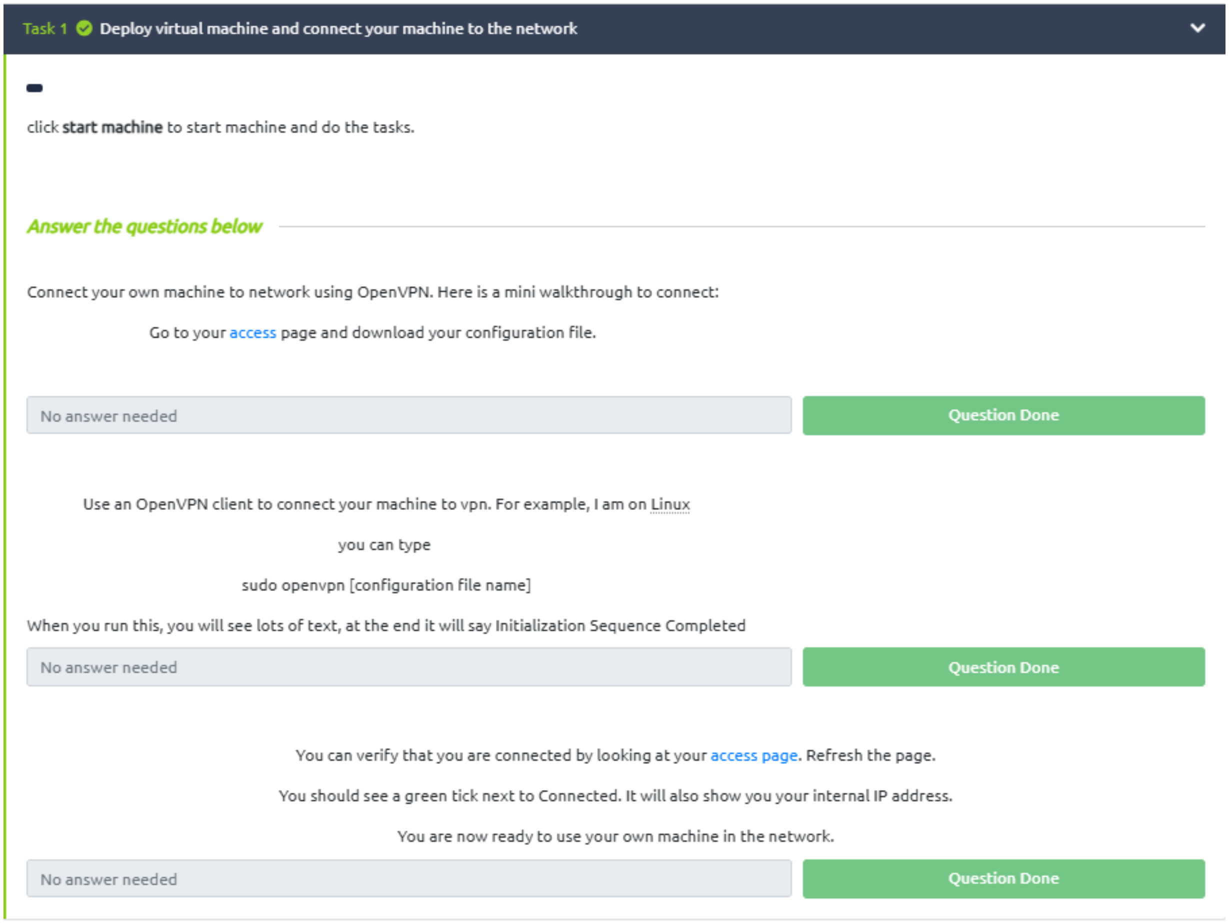
First of all, I was able to successfully connect my computer to the tryhackme network. I created a connection using the provided OpenVPN configuration file, verified its functionality by testing it in a different room, and established the connection. This accomplishment enables me to actively engage in practical training and challenges inside the tryhackme environment.



OpenVPN Access Details

VPN Server Name	IN-Regular-1
Server Status	✓
Connected	✓
Internal Virtual IP Address	10.17.46.196

I also finished creating the first task, which describes how to join a PC to the tryhackme network step-by-step. This job is a great resource, especially for new users who might need help setting up their connection on the tryhackme network.



Task 1 ✓ Deploy virtual machine and connect your machine to the network

click **start machine** to start machine and do the tasks.

Answer the questions below

Connect your own machine to network using OpenVPN. Here is a mini walkthrough to connect:

Go to your access page and download your configuration file.

No answer needed | Question Done

Use an OpenVPN client to connect your machine to vpn. For example, I am on Linux

you can type

```
sudo openvpn [configuration file name]
```

When you run this, you will see lots of text, at the end it will say Initialization Sequence Completed

No answer needed | Question Done

You can verify that you are connected by looking at your access page. Refresh the page.

You should see a green tick next to Connected. It will also show you your internal IP address.

You are now ready to use your own machine in the network.

No answer needed | Question Done

In addition, I successfully created a task that focused on teaching the fundamentals of using the Nmap program. Considering the simple challenge of collecting relevant and credible information, I relied on my basic knowledge of Nmap and the StationX Nmap cheat sheet.

Challenges:

Finding suitable information to develop my Nmap task was one of the challenges I faced. However, using my minimal understanding of Nmap and the Nmap cheat sheet, I was able to solve this challenge. The issue was not insurmountable, and it provided me with an opportunity to further expand my understanding of Nmap and its possibilities.

Resources and References:

https://www.stationx.net/nmap-cheat-sheet/

https://nmap.org/book/man.html

https://www.varonis.com/blog/nmap-commands

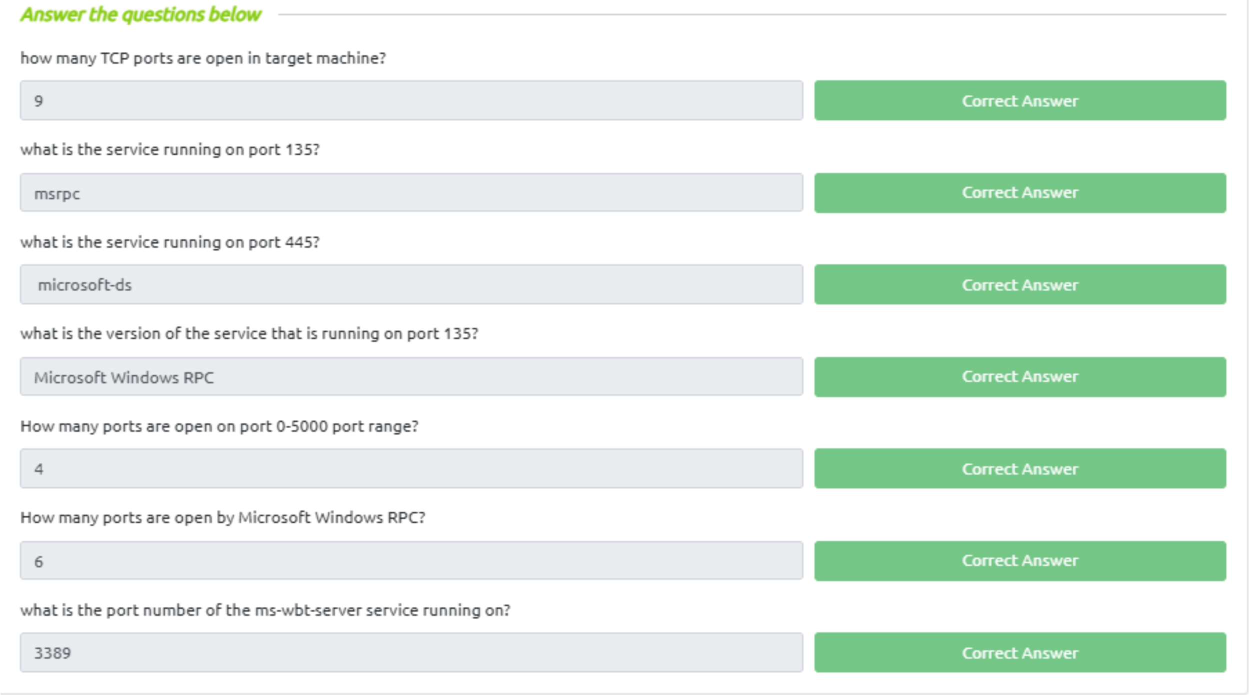
Date: may 15 2023

Today's Goals:

Learn the Python-Nmap library.

Begin my next task using Python-Nmap.

Create unique scanning tasks using Nmap on my cisco-blue virtual machine (VM).



Answer the questions below

how many TCP ports are open in target machine?

9

what is the service running on port 135?

msrpc

what is the service running on port 445?

microsoft-ds

what is the version of the service that is running on port 135?

Microsoft Windows RPC

How many ports are open on port 0-5000 port range?

4

How many ports are open by Microsoft Windows RPC?

6

what is the port number of the ms-wbt-server service running on?

3389

Goals Accomplished:

First, I spent time learning the Python-Nmap library thoroughly. I obtained a basic understanding of its operations and learnt how to use its features for network scanning by using internet resources and documentation. This understanding set the foundation for my next duty and allowed me to comfortably continue my exploration.

Following that, I began working on my next task, which included using Python-Nmap to perform network scans on my personal virtual machine. I configured my VM to accept SSH connections, which would be required for future operations. I also installed Samba on the VM in order to produce a unique question for my scanning experiment.

Challenges:

Opening ports on my virtual system was an interesting problem I experienced during this session. To remedy this, I established an SSH server to ensure that I could access the VM remotely for future operations. I also installed Samba in order to establish a specific environment for the scanning experiment. These steps required careful preparation, but they were ultimately successful in capturing the challenge.

Today's session was both productive and rewarding. I completed my objectives of being acquainted with the Python-Nmap module, starting my next work, and configuring my virtual machine for customized scanning tasks. In the coming days, I plan to delve deeper into Python-Nmap's capabilities and broaden my knowledge of cyber security.

Resources and References:

https://www.geeksforgeeks.org/port-scanner-using-python-nmap/

https://www.studytonight.com/network-programming-in-python/integrating-port-scanner-with-nmap

Date: may 16 2023

Today's Goals:

Learn to obtain properly formatted output from Python-Nmap.

Add a task to teach the most frequently used keys in the inner dictionary of Python-Nmap.

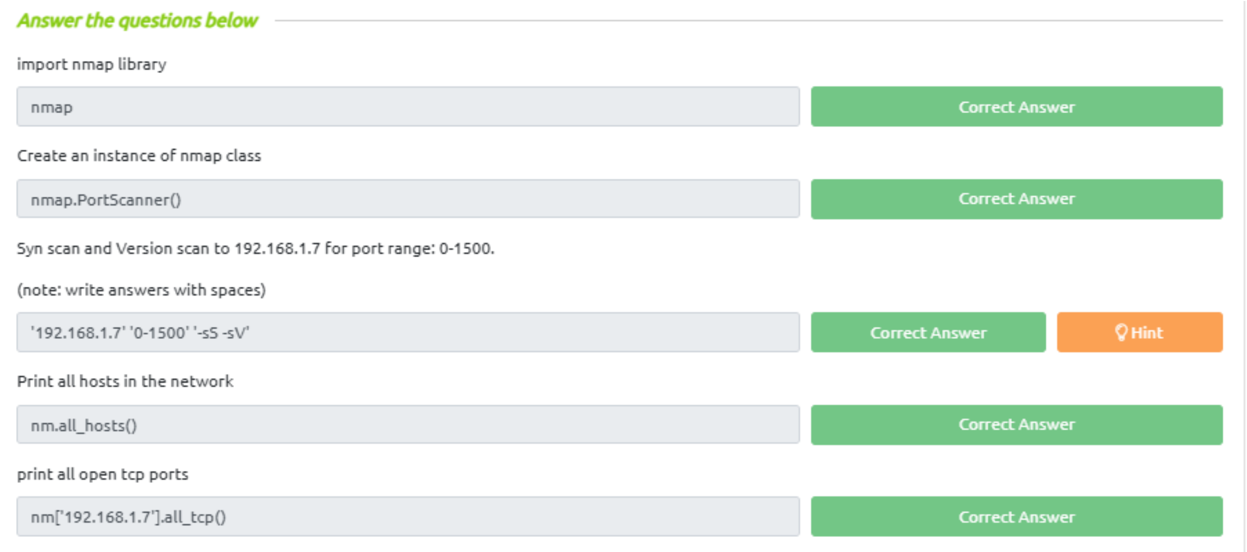
Create practical questions related to the task to enhance understanding.

Goals Accomplished:

I am pleased to note that all of today's objectives are accomplished.

Challenges:

I focused on teaching a simple Python program for network scanning in yesterday's tasks. However, I noticed that the generated output was not as informative as I had hoped. To overcome this issue, I looked for resources to learn how to get more detailed output from Python-Nmap. Unfortunately, obtaining appropriate materials proved tough. As a result, I requested assistance from ChatGPT, which gave me a useful article. I learnt new strategies from the essay and used them to improve the output formatting in the network scanning assignment, which I titled "Network Scanning with Python 2."



Answer the questions below

import nmap library

nmap

Create an instance of nmap class

nmap.PortScanner()

Syn scan and Version scan to 192.168.1.7 for port range: 0-1500.

(note: write answers with spaces)

'192.168.1.7' '0-1500' '-sS -sV'

Print all hosts in the network

nm.all_hosts()

print all open tcp ports

nm['192.168.1.7'].all_tcp()

Another difficulty I experienced was related to the criteria of our task. According to the assignment criteria, students must create a Python program for network scanning. However, it became clear that building a Python script for network scanning was not the most efficient technique. Because the Nmap program is widely available and frequently utilized, it made more sense for people to use it instead. To address this problem, I enlarged the exercise by including a Python script for network scanning as well as questions requiring understanding and analysis of the code.

Resources and References:

https://stackoverflow.com/questions/31104766/parsing-an-nmap-result

https://www.freecodecamp.org/learn/information-security/python-for-penetration-testing/developing-an-nmap-scanner-part-1

https://www.pythonpool.com/python-nmap/

https://medium.com/@toxic.foundersorg/using-the-nmap-port-scanner-with-python-383e8ea47884

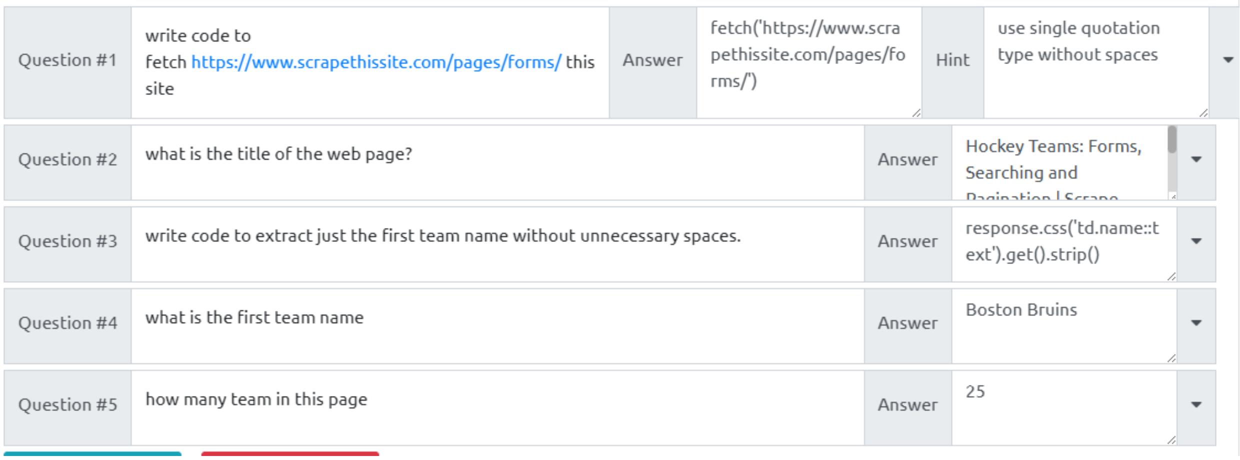
Date: may 17 2023

Today's Goals:

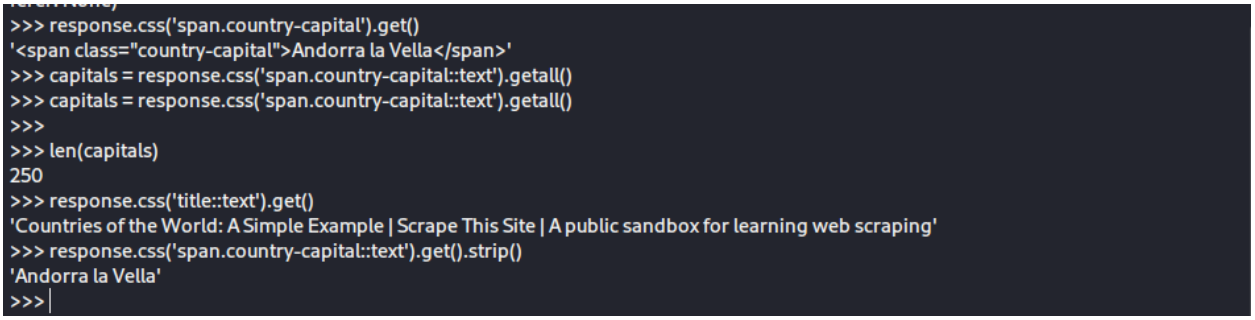
My main goal for today was to start learning Scrapy, a strong Python framework for web crawling. I wanted to learn about Scrapy's features and how it can be used to successfully extract data from websites.

Goals Accomplished:

I successfully began my Scrapy journey by going into its documentation and studying the concept of Scrapy selectors. I learned a lot about selectors and their importance in getting specific data from web pages. Understanding selectors lay the foundation for my future Scrapy web crawling activities.



Question #1	write code to fetch https://www.scrapethissite.com/pages/forms/ this site	Answer	fetch('https://www.scrapethissite.com/pages/forms/')	Hint	use single quotation type without spaces
Question #2	what is the title of the web page?	Answer	Hockey Teams: Forms, Searching and Pagination | Scrape		
Question #3	write code to extract just the first team name without unnecessary spaces.	Answer	response.css('td.name::text').get().strip()		
Question #4	what is the first team name	Answer	Boston Bruins		
Question #5	how many team in this page	Answer	25		



```
>>> response.css('span.country-capital').get()
'<span class="country-capital">Andorra la Vella</span>'
>>> capitals = response.css('span.country-capital::text').getall()
>>> capitals = response.css('span.country-capital::text').getall()
>>>
>>> len(capitals)
250
>>> response.css('title::text').get()
'Countries of the World: A Simple Example | Scrape This Site | A public sandbox for learning web scraping'
>>> response.css('span.country-capital::text').get().strip()
'Andorra la Vella'
>>>
```

Challenges:

I faced a huge challenge during my studies. I realized I couldn't get right into Python code for site crawling without first learning about Scrapy selectors. To overcome this challenge, I shifted my attention to understanding selectors and how they are used in the Scrapy framework. This turn allowed me to lay a solid foundation and ensure that I was prepared to move on to more advance Scrapy concepts.

Furthermore, I came across articles detailing probable installation difficulties, notably due to package incompatibilities. To overcome this difficulty, I chose to install Scrapy in a virtual environment. This method contributed to the creation of an isolated environment, ensuring a seamless installation and reducing conflicts with other packages or dependencies.

Resources and References:

https://docs.scrapy.org/en/latest/intro/tutorial.html

https://www.datacamp.com/tutorial/making-web-crawlers-scrapy-python

https://docs.scrapy.org/en/latest/topics/selectors.html

https://www.geeksforgeeks.org/scrapy-selectors/

Date: may 18 2023

Today's Goals:

Learn to write a Python spider for Scrapy.

Goals Accomplished:

Today, I dedicated my time with Scrapy researching the world of web scraping and successfully completing my goal of learning how to write a spider in Python.

In the context of Scrapy, a spider is a program that navigates webpages, extracts data, and performs various tasks. I obtained a clear knowledge of what a spider is and how it performs within the Scrapy framework through my research and experimentation.

```
import scrapy

class countrySpider(scrapy.Spider):

    name = 'country'

    start_urls = ['https://www.scrapethissite.com/pages/simple/']

    def parse(self, response):

        for country in response.css('div.col-md-4.country'):

            yield {

                'capital': teams.css('span.country-capital::text').get(),

                'population': teams.css('span.country-population::text').get(),

                'area' : teams.css('span.country-area::text').get(),

            }

                next_page = response.css('li.next_page a::attr(href)').get()

               if next_page is not None:

                        yield response.follow(next_page, self.parse)
```

Challenges:

During the process, I ran into a problem gathering data from all of the web pages I meant to scrape. However, I quickly found a solution by utilizing the functionality of the website's "next" button. I was able to browse through the web sites systematically by implementing the appropriate logic within my spider, ensuring that I extracted data from all of the desired pages.

Overcoming this challenge not only showed me the value of problem-solving in online scraping, but it also presented me with a valuable technique for ensuring full data extraction from multi-page websites.

Resources and References:

https://www.youtube.com/watch?v=s4jtkzHhLzY&t=240s

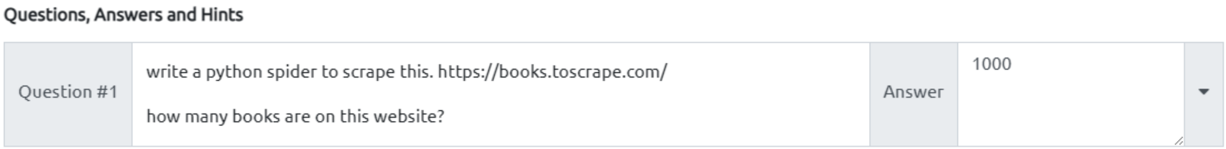
https://docs.scrapy.org/en/latest/topics/spiders.html

Date: may 19 2023

Today's Goals:

Create a Task for Python Scrapy Installation in TryHackMe Room

create another task to teach python programming to write a sipider in scrapy



Questions, Answers and Hints

Question #1	write a python spider to scrape this. https://books.toscrape.com/ how many books are on this website?	Answer	1000

Goals Accomplished:

I am delighted to report that I accomplished both duties satisfactorily. I gave step-by-step directions for installing Scrapy, including the required dependencies and virtual environment configuration, for the installation operation. In addition, I included useful resources and troubleshooting suggestions to guarantee that users' installation run successfully.

In the second work, I learned the fundamentals of Python programming for web scraping. I went over key fundamentals including choosing and parsing HTML components, traversing online pages, and extracting necessary data with Scrapy. I gave people concrete examples and encouraged them to build their own spiders to scrape certain pages, improving direct learning and exploration.

Challenges:

I did not face any big difficulties while developing these tasks. Fortunately, I had prior knowledge and expertise with Python Scrapy, which helped me to properly organize and create the jobs. This prior knowledge reduced any challenges I may have encountered during the creation process, allowing me to focus on providing users with clear and straightforward instructions.

Date: may 20 2023

Today's Goals:

My main goals for today were to learn about file security principles like RelRO (Relocation Read-Only), canaries, NX (No Execute), and PIE (Position Independent Executable)

I wanted to learn about buffer overflow vulnerabilities and how to use the powerful tool Pwntools for exploit development.

Goals Accomplished:

By getting a basic knowledge of file security concepts, I made progress toward my objectives. I investigated the role of features such as RelRO, canaries, NX, and PIE in mitigating different kinds of attacks. While I still have a lot to learn in this area, I now have a solid base to work from.

In terms of buffer overflow, I looked deep into the subject and learned the principles of how these flaws may be exploited. I learned about the potential dangers and repercussions of buffer overflows, which will help me identify and address such vulnerabilities in the future.

Challenges:

The complexity of exploit development was one of the most difficult challenges I faced during this process of learning. Exploit development is a highly specialized talent that requires an in-depth knowledge of low-level programming and system flaws. It became clear that simply learning Python exploit development without having a deeper base would be insufficient.

Moving forward, I acknowledge the need to improve my understanding of system architecture, memory management, and assembly language in order to pursue exploit development efficiently.

Resources and References:

https://tc.gts3.org/cs6265/2019/tut/tut03-02-pwntools.html

https://github.com/revanmalang/pwntools

https://tryhackme.com/room/introtopwntools

Date: may 21 2023

Today's Goals:

Create a vulnerable C program that caters to the flag using the "gets" function.

Compile the C program with security measures using GCC.

Compile another version of the program without security using specific flags (-m32 -fno-stack-protector -z execstack) and setuid permissions to make it less secure.

Add a new user to the TryHackMe platform.

Generate a flag using the newly added user and set the permissions to read and write only for the owner and group.

Install Pwntools, a Python library for exploit development.

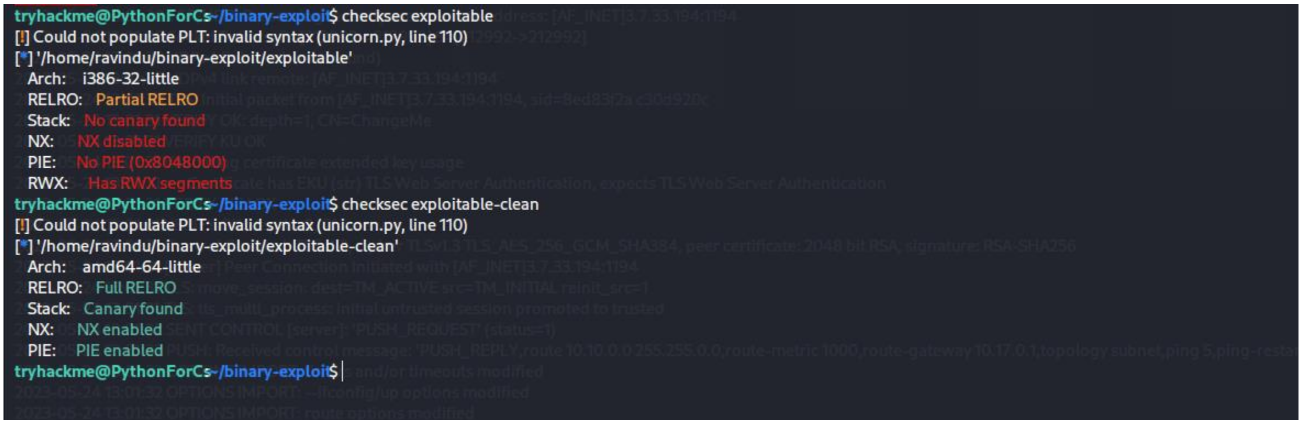
Install Pwngdb to facilitate the debugging of the program.

Remove sudo permissions from the TryHackMe user.

Successfully upload the finalized VM to the TryHackMe platform

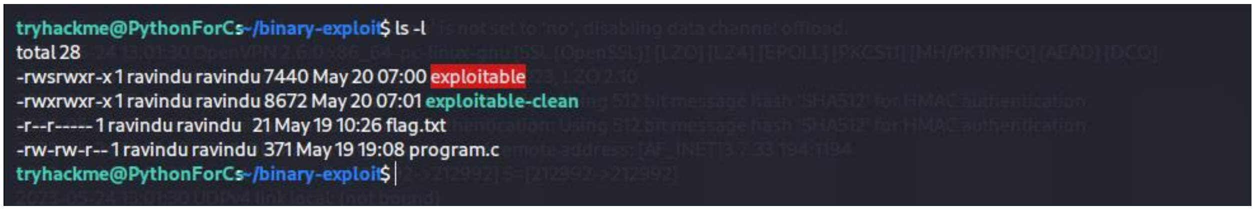
Goals Accomplished:

I made tremendous progress on my goals. I successfully built a vulnerable C program that uses the "gets" function to retrieve the flag. In addition, I used GCC to generate the software with security features. In addition, I created a copy of the application without security protections, using certain flags (--fno-stack-protector --no-pie) and setuid permissions to reduce its security level.

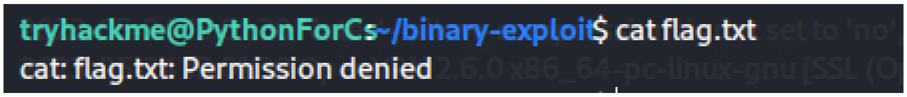


```
tryhackme@PythonForCs~/binary-exploit$ checksec exploitable
[!] Could not populate PLT: invalid syntax (unicorn.py, line 110)
[*] '/home/ravindu/binary-exploit/exploitable'
    Arch:     i386-32-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX disabled
    PIE:      No PIE (0x8048000)
    RWX:      Has RWX segments
tryhackme@PythonForCs~/binary-exploit$ checksec exploitable-clean
[!] Could not populate PLT: invalid syntax (unicorn.py, line 110)
[*] '/home/ravindu/binary-exploit/exploitable-clean'
    Arch:     amd64-64-little
    RELRO:    Full RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
tryhackme@PythonForCs~/binary-exploit$
```

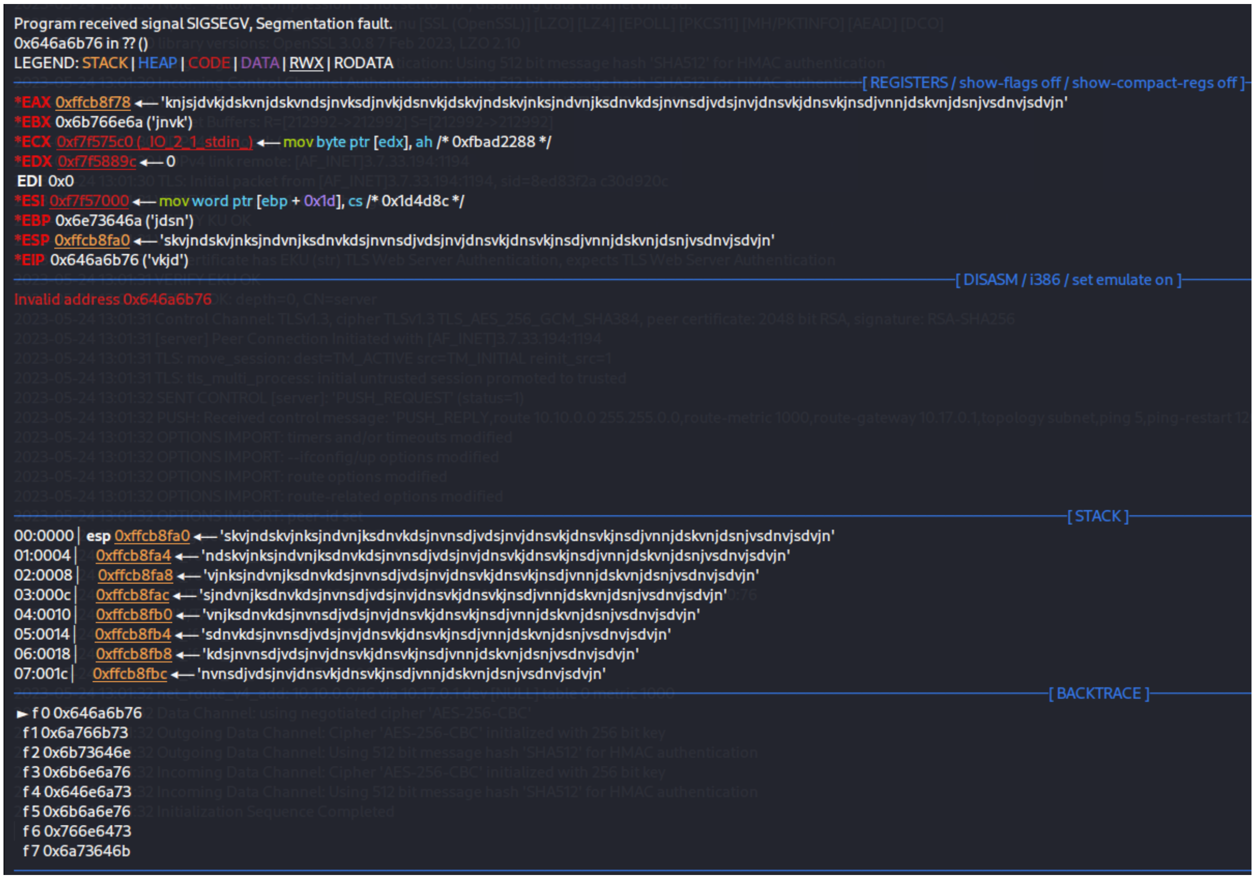
I added a new user to the TryHackMe platform to improve the CTF experience and generated a flag that only the owner and group can read and write. I also installed Pwntools and Pwngdb, which will help with exploit development and software debugging. Finally, I deleted the TryHackMe user's sudo capabilities to ensure the intended challenge complexity.



```
tryhackme@PythonForCs~/binary-exploit$ ls -l
total 28
-rwsrwxr-x 1 ravindu ravindu 7440 May 20 07:00 exploitable
-rwxrwxr-x 1 ravindu ravindu 8672 May 20 07:01 exploitable-clean
-r--r----- 1 ravindu ravindu   21 May 19 10:26 flag.txt
-rw-rw-r-- 1 ravindu ravindu  371 May 19 19:08 program.c
tryhackme@PythonForCs~/binary-exploit$ |
```



```
tryhackme@PythonForCs~/binary-exploit$ cat flag.txt
cat: flag.txt: Permission denied
```



```
Program received signal SIGSEGV, Segmentation fault.
0x646a6b76 in ?? ()
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
──────────────────────────────────────────[ REGISTERS / show-flags off / show-compact-regs off ]─
*EAX  0xffcb8f78 ◂— 'knjsjdvkjdskvnjdskvndsjnvksdjnvkjdsnvkjdskvjndskvjnksjndvnjksdnvkdsjnvnsdjvdsjnvjdnsvkjdnsvkjnsdjvnnjdskvnjdsnjvsdnvjsdvjn'
*EBX  0x6b766e6a ('jnvk')
*ECX  0xf7f575c0 (_IO_2_1_stdin_) ◂— mov byte ptr [edx], ah /* 0xfbad2288 */
*EDX  0xf7f5889c ◂— 0
 EDI  0x0
*ESI  0xf7f57000 ◂— mov word ptr [ebp + 0x1d], cs /* 0x1d4d8c */
*EBP  0x6e73646a ('jdsn')
*ESP  0xffcb8fa0 ◂— 'skvjndskvjnksjndvnjksdnvkdsjnvnsdjvdsjnvjdnsvkjdnsvkjnsdjvnnjdskvnjdsnjvsdnvjsdvjn'
*EIP  0x646a6b76 ('vkjd')
─────────────────────────────────────────────────────────[ DISASM / i386 / set emulate on ]──────
Invalid address 0x646a6b76









─────────────────────────────────────────────────────────────────────[ STACK ]────────────
00:0000│ esp 0xffcb8fa0 ◂— 'skvjndskvjnksjndvnjksdnvkdsjnvnsdjvdsjnvjdnsvkjdnsvkjnsdjvnnjdskvnjdsnjvsdnvjsdvjn'
01:0004│     0xffcb8fa4 ◂— 'ndskvjnksjndvnjksdnvkdsjnvnsdjvdsjnvjdnsvkjdnsvkjnsdjvnnjdskvnjdsnjvsdnvjsdvjn'
02:0008│     0xffcb8fa8 ◂— 'vjnksjndvnjksdnvkdsjnvnsdjvdsjnvjdnsvkjdnsvkjnsdjvnnjdskvnjdsnjvsdnvjsdvjn'
03:000c│     0xffcb8fac ◂— 'sjndvnjksdnvkdsjnvnsdjvdsjnvjdnsvkjdnsvkjnsdjvnnjdskvnjdsnjvsdnvjsdvjn'
04:0010│     0xffcb8fb0 ◂— 'vnjksdnvkdsjnvnsdjvdsjnvjdnsvkjdnsvkjnsdjvnnjdskvnjdsnjvsdnvjsdvjn'
05:0014│     0xffcb8fb4 ◂— 'sdnvkdsjnvnsdjvdsjnvjdnsvkjdnsvkjnsdjvnnjdskvnjdsnjvsdnvjsdvjn'
06:0018│     0xffcb8fb8 ◂— 'kdsjnvnsdjvdsjnvjdnsvkjdnsvkjnsdjvnnjdskvnjdsnjvsdnvjsdvjn'
07:001c│     0xffcb8fbc ◂— 'nvnsdjvdsjnvjdnsvkjdnsvkjnsdjvnnjdskvnjdsnjvsdnvjsdvjn'
───────────────────────────────────────────────────────────────────[ BACKTRACE ]──────────
 ► f 0 0x646a6b76
   f 1 0x6a766b73
   f 2 0x6b73646e
   f 3 0x6b6e6a76
   f 4 0x646e6a73
   f 5 0x6b6a6e76
   f 6 0x766e6473
   f 7 0x6a73646b
──────────────────────────────────────────────────────────────────────────────────────────
```

Challenges:

Throughout the day, I experienced a few problems. One of the major challenges was that the C program had built-in security features when I first compiled it. I had to experiment with several approaches to eliminate these security elements while keeping the vulnerability intact. Another issue developed during the testing of the exploit, which resulted in a "permission denied" error. I used setuid permissions on the vulnerable file to get around this.

In addition, I experienced a delay in uploading the VM to TryHackMe, as the procedure appeared to be taking longer than expected. Despite this challenge, I stayed patient and monitored the process.

Resources and References:

https://www.tenouk.com/Bufferoverflowc/Bufferoverflow6.html

https://www.geeksforgeeks.org/buffer-overflow-attack-with-example/

https://www.fortinet.com/resources/cyberglossary/buffer-overflow

https://opensource.com/article/21/6/linux-checksec

https://docs.pwntools.com/en/stable/util/cyclic.html

https://medium.com/@two06/solving-a-simple-buffer-overflow-with-pwntools-575a37e4ddb1

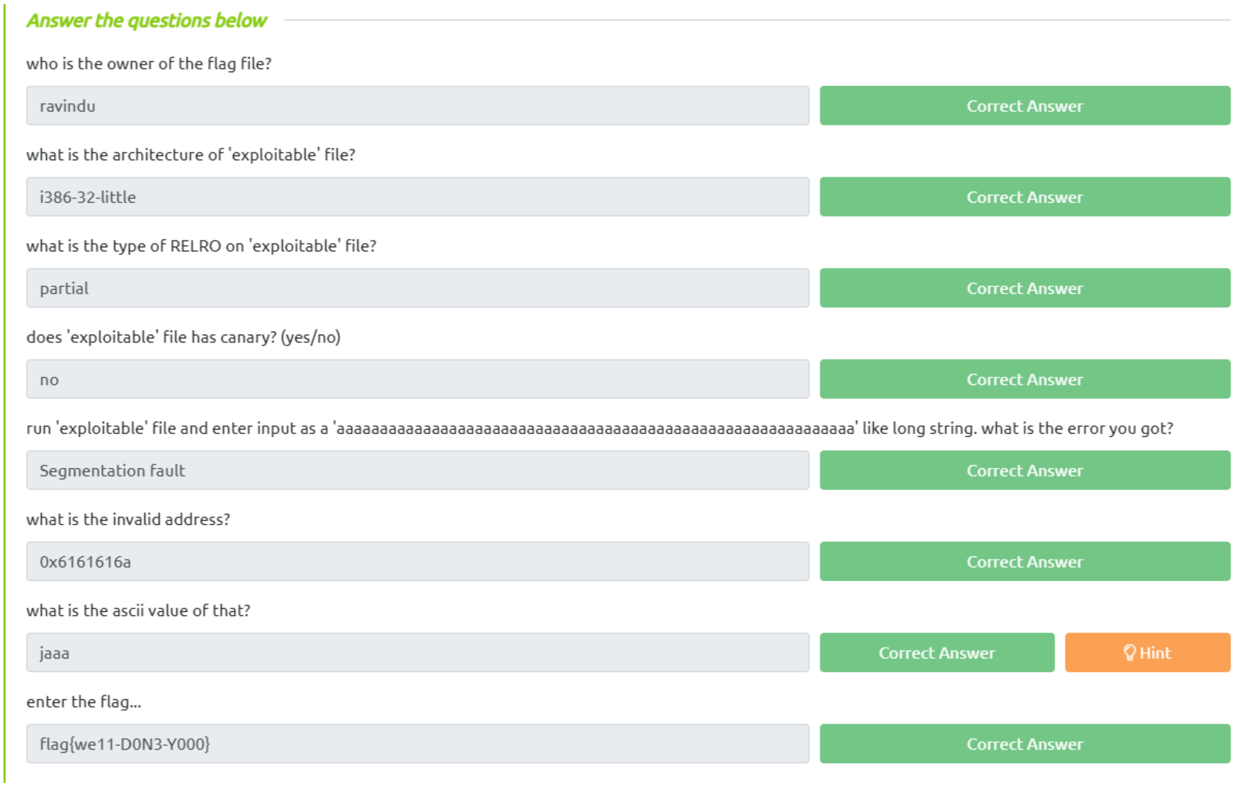
https://www.youtube.com/watch?v=9wepzpQhhio&t=968s

https://www.youtube.com/watch?v=21GW7xQqMqg

Date: may 23 2023

Today's Goals:

Today, I focused on implementing exploit development activities for my Python for Cyber Security assignment on my virtual machine (VM). I set out to perform the prescribed tasks and gain hands-on experience with Python exploit development.



who is the owner of the flag file?

ravindu

what is the architecture of 'exploitable' file?

i386-32-little

what is the type of RELRO on 'exploitable' file?

partial

does 'exploitable' file has canary? (yes/no)

no

run 'exploitable' file and enter input as a 'aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa' like long string. what is the error you got?

Segmentation fault

what is the invalid address?

0x6161616a

what is the ascii value of that?

jaaa

enter the flag...

flag{we11-D0N3-Y000}

Goals Accomplished:

I completed all of the exploit development chores on my VM. I was able to create powerful exploits and generate payloads by utilizing my Python skills and the Pwntools module. This achievement is a crucial milestone in my effort to improve my cyber security skills.

Challenges:

However, I came across a problem during this process. Unfortunately, my VM was still in the process of getting converting, which stopped me from accessing the essential environment for completing the tasks. Even submitting a help ticket, I have yet to receive an answer. I looked into the TryHackMe Discord community for a solution and noticed that the staff is now unavailable due to an ongoing event. This situation has caused delays and impacted my efforts to complete the task on time.