Used exploit-db.com | configured **exploit script** to execute RCE on target machine

Exploit Vulnerabilities

Premium room

Learn about some of the tools, techniques and resources to exploit vulnerabilities

Task 1 Introduction

In this room, we are going to be going over some means of identifying vulnerabilities and coupling our research skills to learn how these can be abused.

Additionally, you will find some publicly available resources that are essential additions to your skill set and tools when performing vulnerability research and exploitation. You will then get to apply all of this into a practical challenge at the end of the room.

Answer the questions below

Let's proceed.

No answer needed

Task 2 Automated Vs. Manual Vulnerability Research

There is a myriad of tools and services available in cybersecurity for vulnerability scanning. Ranging from being commercial (and footing a heavy bill) to open-source and free, vulnerability scanners are convenient means of quickly canvassing an application for flaws.

Nessus® vulnerability scanner

For example, the vulnerability scanner Nessus has both a free (community) edition and commercial. The commercial version costing thousands of pounds for a year's license will likely be used in organisations providing penetration testing services or audits. If you'd like to know more about Nessus, check out the TryHackMe room dedicated to it.

I have detailed some of the advantages and disadvantages of using a vulnerability scanner in the table below:

Advantage	Disadvantage
Automated scans are easy to repeat, and the results can be shared within a team with ease.	People can often become reliant on these tools.
These scanners are quick and can test numerous applications efficiently.	They are extremely "loud" and produce a lot of traffic and logging. This is not good if you are trying to bypass firewalls and the likes.
Open-source solutions exist.	Open-source solutions are often basic and require expensive licenses to have useful features.
Automated scanners cover a wide range of different vulnerabilities that may be hard to manually search for.	They often do not find every vulnerability in an application.

Frameworks such as Metasploit often have vulnerability scanners for some modules; this is something you will come onto learn about in a further module in this pathway.

Manual scanning for vulnerabilities is often the weapon of choice by a penetration tester when testing individual applications or programs. In fact, manual scanning will involve searching for the same vulnerabilities and uses similar techniques as automated scanning.

Ultimately, both techniques involve testing an application or program for vulnerabilities. These vulnerabilities include:

Vulnerability	Description
Security Misconfigurations	Security misconfigurations involve vulnerabilities that are due to developer oversight. For example, exposing server information in messages between the application and an attacker.
Broken Access Control	This vulnerability occurs when an attacker is able to access parts of an application that they are not supposed to be able to otherwise.
Insecure Deserialization	This is the insecure processing of data that is sent across an application. An attacker may be able to pass malicious code to the application, where it will then be executed.
Injection	An Injection vulnerability exists when an attacker is able to input malicious data into an application. This is due to the failure of not ensuring (known as sanitising) input is not harmful.

If you are keen to learn more about these vulnerabilities, the OWASP framework will be a useful read to you. TryHackMe even has a room showcasing the top ten vulnerabilities outlined by OWASP.

Answer the questions below

You are working close to a deadline for your penetration test and need to scan a web application quickly. Would you use an automated scanner? (Yay/Nay)

Yay

You are testing a web application and find that you are able to input and retrieve data in a database. What vulnerability is this?

Injection

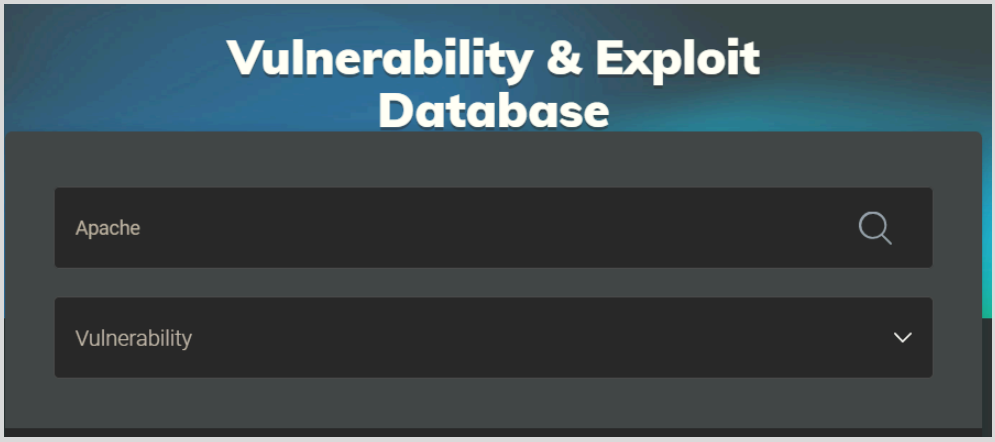
You manage to impersonate another user. What vulnerability is this?

Broken Access Control

Task 3 Finding Manual Exploits

Rapid7

Much like other services such as Exploit DB and NVD, Rapid7 is a vulnerability research database. The only difference being that this database also acts as an exploit database. Using this service, you can filter by type of vulnerability (I.e. application and operating system).



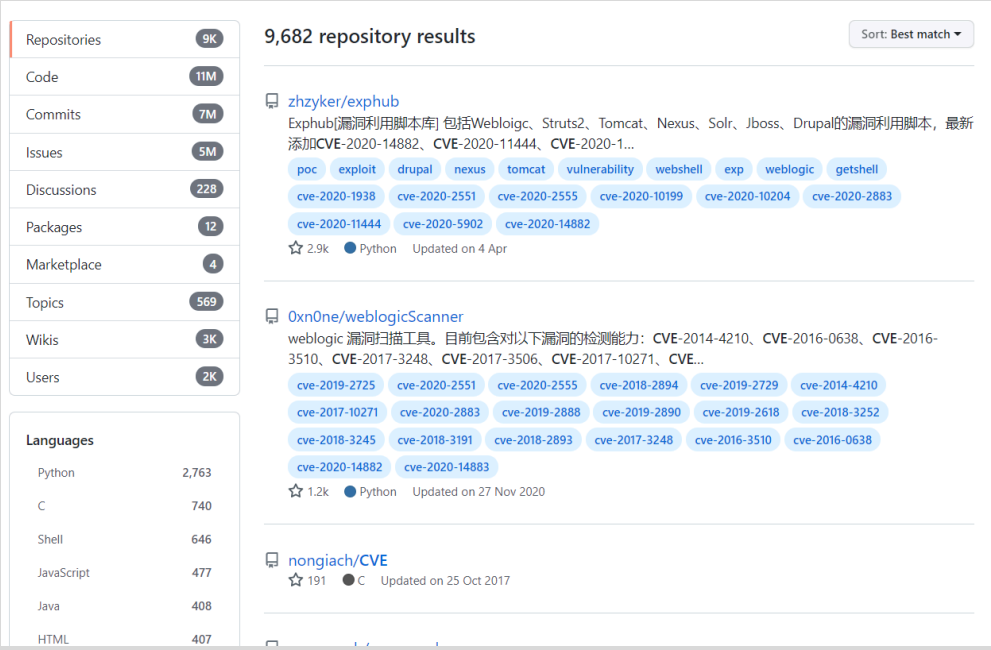
Additionally, the database contains instructions for exploiting applications using the popular Metasploit tool (you will learn about this tool in-depth later in the learning path). For example, this entry on Rapid7 is for “Wordpress Plugin SP Project & Document”, where we can see instructions on how to use an exploit module to abuse this vulnerability.

```
1  msf > use exploit/multi/http/wp_plugin_sp_project_document_rce
2  msf exploit(wp_plugin_sp_project_document_rce) > show targets
3      ...targets...
4  msf exploit(wp_plugin_sp_project_document_rce) > set TARGET < target-id >
5  msf exploit(wp_plugin_sp_project_document_rce) > show options
6      ...show and set options...
7  msf exploit(wp_plugin_sp_project_document_rce) > exploit
```

GitHub

GitHub is a popular web service designed for software developers. The site is used to host and share the source code of applications to allow a collaborative effort. However, security researchers have taken to this platform because of the aforementioned reasons as well. Security researchers store & share PoC's (Proof of Concept) on GitHub, turning it into an exploit database in this context.

GitHub is extremely useful in finding rare or fresh exploits because anyone can create an account and upload – there is no formal verification process like there is with alternative exploit databases. With that said, there is also a downside in that PoC's may not work where little to no support will be provided.



GitHub uses a tagging and keyword system, meaning that we can search GitHub by keywords such as "PoC", "vulnerability", and many more. At the time of writing, there are 9,682 repositories with the keyword "cve". We are also able to filter the results by programming language.

Searchsploit

Searchsploit is a tool that is available on popular pentesting distributions such as Kali Linux. It is also available on the TryHackMe AttackBox. This tool is an offline copy of Exploit-DB, containing copies of exploits on your system.

You are able to search searchsploit by application name and/or vulnerability type. For example, in the snippet below, we are searching searchsploit for exploits relating to Wordpress that we can use – no downloading necessary!

Using Searchsploit to look for exploits relating to "Wordpress"

```
searchsploit wordpress
WordPress Theme Think Responsive 1.0 - Arbitr | php/webapps/29332.txt
```

```
WordPress Theme This Way - 'upload_settings_i | php/webapps/38820.php
WordPress Theme Toolbox - 'mls' SQL Injection | php/webapps/38077.txt
WordPress Theme Trending 0.1 - 'cpage' Cross- | php/webapps/36195.txt
WordPress Theme Uncode 1.3.1 - Arbitrary File | php/webapps/39895.php
WordPress Theme Urban City - 'download.php' A | php/webapps/39296.txt
WordPress Theme Web Minimalist 1.1 - 'index.p | php/webapps/36184.txt
WordPress Theme White-Label Framework 2.0.6 - | php/webapps/38105.txt
WordPress Theme Wp-ImageZoom - 'id' SQL Injec | php/webapps/38063.txt
WordPress Theme Zoner Real Estate - 4.1.1 Per | php/webapps/47436.txt
```

Answer the questions below

What website would you use as a security researcher if you wanted to upload a Proof of Concept?

Github

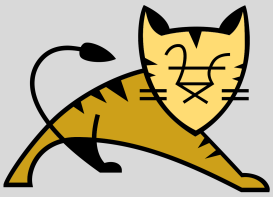
You are performing a penetration test at a site with no internet connection. What tool could you use to find exploits to use?

Searchsploit

Task 4 Example of Manual Exploitation

We can use the information gathered from task 2 in this room to exploit the vulnerable service. Ultimately, one of the most effective vulnerabilities that we can exploit is the ability to execute commands on the target that is running the vulnerable application or service.

For example, being able to execute commands on the target that is running the vulnerable application or service will allow us to read files or execute commands that we previously wouldn't be able to perform using the application or service alone. Additionally, we can abuse this to gain what is known as a foothold to the machine. A foothold is an access to the vulnerable machine's console, where we can then begin to exploit other applications or machines on the network.



Apache Tomcat

We are going to use an exploit to perform remote code execution on the application from task 2 to be able to remotely execute commands on the vulnerable machine.

Before we start, it is important to note that exploits rarely come out of the box and are ready to be used. They often require some configuration before they will work for our environment or target. The level of configuration will vary upon the exploit, so you will often find multiple exploits for the same vulnerability on an application. It is up to you to figure out which exploit is the most appropriate or useful to you.

For example, in the snippet below, we can see that a few options have been changed to reflect the IP address of the machine that we are attacking from.

Modifying an Exploit (Before)

```
nano exploit.py
mymachine="192.168.1.10"
port="1337"
```

Modifying an Exploit (After)

```
nano exploit.py
mymachine="10.13.37.10"
port="1337"
```

Once we have configured the exploit correctly, let's further read this exploit to understand how to use it. In the snippet below, we can see that we need to provide two arguments when running the exploit:

Listing the arguments for an exploit

```
exploit.py --help
To use this exploit, provide the following arguments:
-u The URL of the application
-c the command that you wish to execute
```

With this information in mind, we are now ready to use this exploit on the vulnerable machine. We are going to do the following:

1. Use the exploit to upload a malicious file to the vulnerable application containing whatever command we wish to execute, where the web server will run this malicious file to execute the code.
2. The file will first contain a basic command that we will use to verify that the exploit has worked.
3. Then we are going to read the contents of a file located on the vulnerable machine.

Running the exploit to output the name of the user that the application is running as

```
exploit.py -u http://10.10.10.10 -c "whoami"
www-data
```

Running the exploit to output the contents of a file on the target machine

```
exploit.py -u http://10.10.10.10 -c "cat flag.txt"
THM{EXPLOIT_COMPLETE}
```

Answer the questions below

What type of vulnerability was used in this attack?

Remote Code Execution

Task 5 Practical: Manual Exploitation

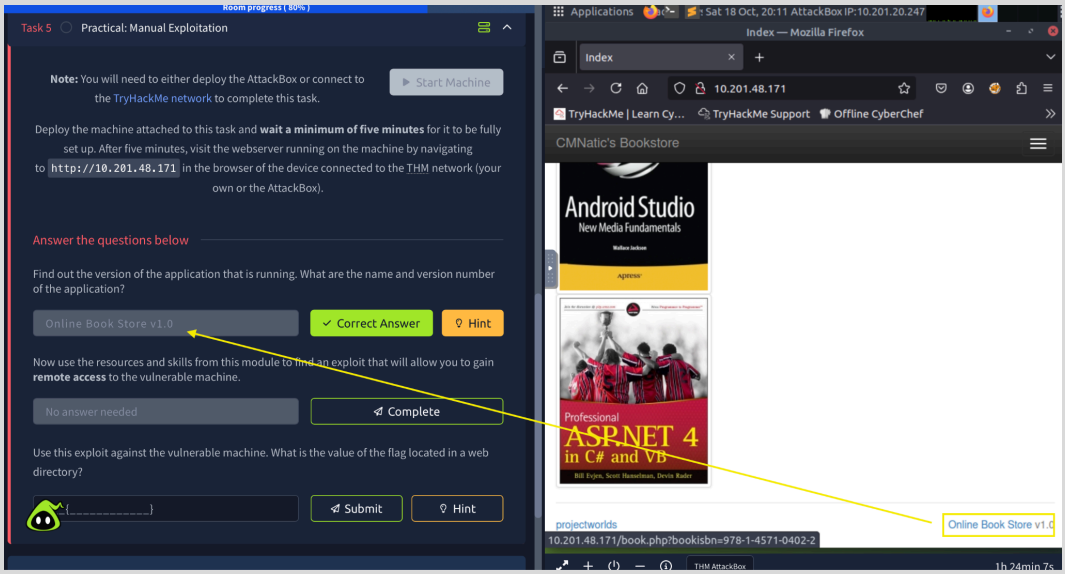
Note: You will need to either deploy the AttackBox or connect to the TryHackMe network to complete this task.

Deploy the machine attached to this task and wait a minimum of five minutes for it to be fully set up. After five minutes, visit the webserver running on the machine by navigating to `http://MACHINE_IP` in the browser of the device connected to the THM network (your own or the AttackBox).

Answer the questions below

Find out the version of the application that is running. What are the name and version number of the application?

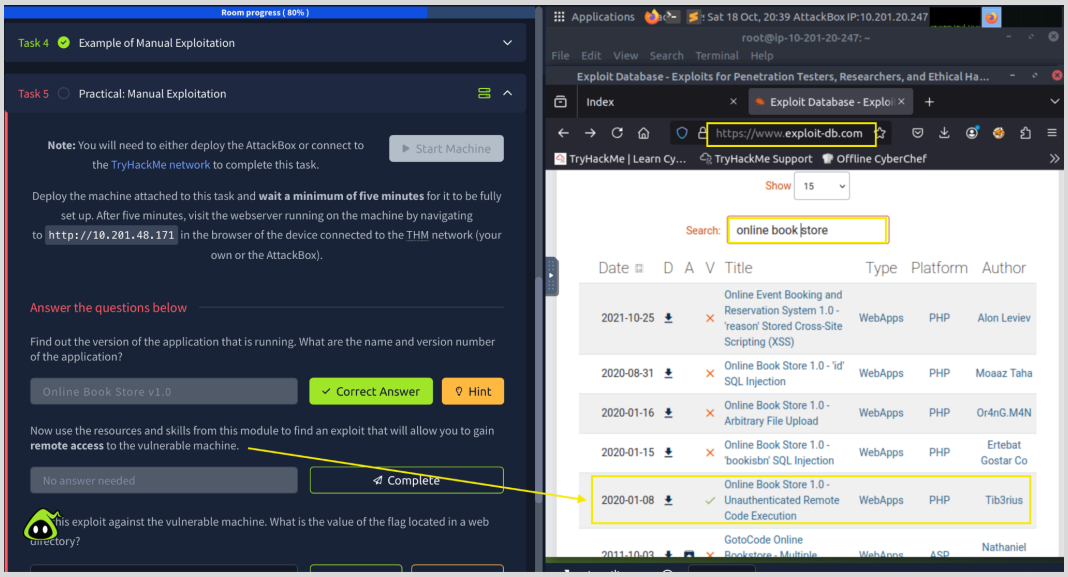
Online Book Store v1.0



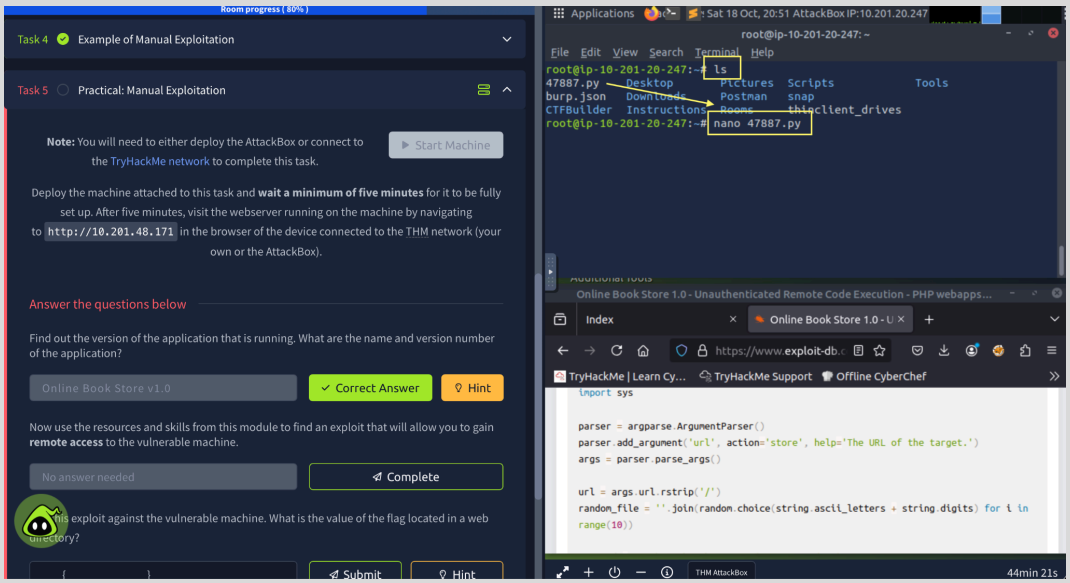
Now use the resources and skills from this module to find an exploit that will allow you to gain **remote access** to the vulnerable machine.

No answer needed

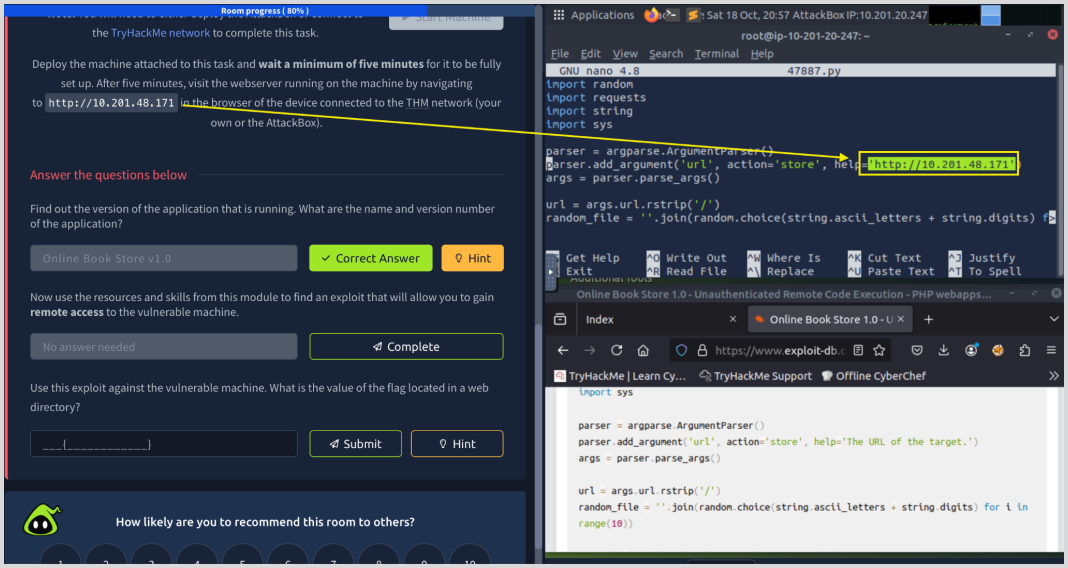
Step 1: Go to: exploit-db.com → Search: 'online book store' → download the exploit



Step 2: Open the Terminal → run 'ls' command to check for the script file to be edited



Step 2: edit the 'help' parameter of the script → save



Use this exploit against the vulnerable machine. What is the value of the flag located in a web directory?

THM{BOOK_KEEPING}