1) Vulnerabilities 101 :-

1. What is a Vulnerability?

In simple terms, a vulnerability is a flaw or a weakness in a system’s design or code. Think of it like a window in a house that was built with a broken lock. It’s not a problem until someone notices it and uses it to get inside.

In cybersecurity, we use three main terms to describe the process of a hack:

Vulnerability: The weakness itself.

Exploit: The specific action or "tool" used to take advantage of that weakness.

Proof of Concept (PoC): A demonstration showing that the weakness can actually be exploited.

The 5 Main Categories of Vulnerabilities

The lab breaks down vulnerabilities into five primary groups based on where the weakness lives:

Operating System (OS): Flaws found directly within the OS itself (like Windows or Linux). These often lead to privilege escalation, where a normal user gains admin-level control.

Misconfiguration-based: These happen when an application or service is set up incorrectly. A common example is a website accidentally leaving customer details exposed to the public because of a wrong setting.

Weak or Default Credentials: This is one of the easiest for attackers to exploit. Many services come with pre-set usernames and passwords (like "admin" / "admin") that users forget to change.

Application Logic: These are bugs caused by poorly designed applications. An attacker might find a way to trick the app's internal "logic" to impersonate another user.

Human-Factor: These vulnerabilities target people rather than code. The most famous example is phishing, where deceptive emails trick people into giving away sensitive information.

Explaining the Lab Answers

Based on the descriptions above, here is why those specific answers were correct:

Question: An attacker has been able to upgrade the permissions of their system account from "user" to "administrator". What type of vulnerability is this?

Answer: Operating System.

Why: The description for OS vulnerabilities explicitly mentions that they often result in privilege escalation (moving from user to admin).

Question: You manage to bypass a login panel using cookies to authenticate. What type of vulnerability is this?

Answer: Application Logic.

Why: This is a case of a poorly implemented authentication mechanism. By using cookies to bypass the normal login flow, you are exploiting a flaw in how the application was designed to handle user access.

3. Scoring the Risk (CVSS vs. VPR)

Since companies can't fix everything at once, they use scores to prioritize.

CVSS: This is the old-school, free standard. It gives a score based on how bad the technical flaw is. It's static, meaning the score doesn't change much over time.

VPR: This is a newer, commercial way (by Tenable). It's dynamic and risk-driven, meaning the score changes daily based on what hackers are actually doing in the real world.

How I got the Lab Answers:

Year of first CVSS: The text says it was introduced in 2005.

Risk-based framework: VPR, because it prioritizes what to patch based on real-world threat levels.

Free and open-source: CVSS, because anyone can use the calculator for free.

4. Finding the Data (NVD & Exploit-DB)

Finally, I looked at where this info is stored.

NVD (National Vulnerability Database): This is the main list for CVEs. A CVE ID looks like CVE-2017-0144—the "2017" is the year it was found.

Exploit-DB: This is where you find the actual code to run an exploit during a test.

Using NVD, how many CVEs were published in July 2021?
- the answer is 1554 to get this visit nvd site and on side window after clicking advance search option in published date add 1st july 2021 to 31 july 2021 and we will get to see total number of vulnerabilities in statistic window

Who is the author of Exploit-DB?
- the author of Exploit DB is offsec when we open its official site it is mentioned below.

5. an example of finding vulnerability

What type of vulnerability did we use to find the name and version of the application in this example?
- the answer is version disclosure as we learned about it in in this task.

6. Showcase: Exploiting Ackme's Application

1. Information Gathering with Nmap

The first thing I did was run an Nmap scan against the lab's machine. Nmap is a powerful tool that scans "ports" to see what services are running.

What I found: The scan showed that a web server was open. More importantly, Nmap identified the specific application and its version: "Online Book Store v1.0".

Why this was the key: In security, knowing the exact version of software is half the battle. If you know the version, you can check if anyone else has already found a "hole" in it.

2. Researching the Exploit

Now that I knew I was looking at Online Book Store v1.0, I checked the vulnerability databases we talked about earlier (like Exploit-DB).

I searched for that specific version and found a major flaw: Unauthenticated Remote Code Execution (RCE).

This is a "Critical" vulnerability because it means I can run commands on the server without even needing a username or password.

3. Executing the Attack and Getting the Flag

I followed the steps in the lab to launch the exploit against the site.

I selected the exploit for Online Book Store 1.0.

I ran the script.

Because the version was outdated and unpatched, the exploit worked!

Once the exploit finished, the system gave me the "Flag," which is the secret code that proves I successfully completed the hack.

The Flag: THM{ACKME_ENGAGEMENT}

Web application securtity:-

This lab was a simple breakdown of how we access information online using a browser. Here is the short version for the report.

What is the Web?

The "Web" (World Wide Web) is basically a huge collection of pages and files stored on computers called Servers. We access these pages using the internet.

What is a Browser?

A Browser (like Chrome, Firefox, or Safari) is the software application we use to view the web. It acts as a translator: it takes the code sent by a server and turns it into the text, images, and videos we see on our screens.

How I Got the Answers

The lab asked a key question to make sure I understood the difference between the internet and the tools we use to see it.

Question: What do you need to access the web application?

Answer: Browser, cause we access web through browsers

2. Web Application Security Assessment

1. Identification and Authentication Failure

Vulnerability Discovered: The login page allowed for unlimited login attempts without implementing rate limiting, account lockouts, or CAPTCHA challenges.

Reasoning: This falls under Identification and Authentication Failure (formerly known as Broken Authentication) because the system fails to protect against automated brute-force attacks. Without a "lockout" mechanism, an attacker can try thousands of password combinations until they gain access.

2. Cryptographic Failures

Vulnerability Discovered: Sensitive data, specifically usernames and passwords, were transmitted in cleartext (without encryption) across the network.

Reasoning: This is categorized as a Cryptographic Failure (formerly Sensitive Data Exposure). When data is sent over HTTP instead of HTTPS, or without proper encryption protocols, it can be intercepted by anyone on the same network (Man-in-the-Middle attack).

3. Injection (SQLi)

Vulnerability Discovered: The application was susceptible to an SQL Injection attack via the login form (using the ' OR 1=1 -- bypass).

Reasoning: This is a classic Injection flaw. Because the application did not sanitize user input, the input was treated as a command by the database. This allowed for unauthorized access by manipulating the SQL query to always return "True."

4. Broken Access Control

Vulnerability Discovered: Accessing sensitive directories (like /admin) simply by guessing the URL or finding it via automated tools like gobuster.

Reasoning: This represents Broken Access Control. The application relied on "security by obscurity" rather than enforcing strict permissions. If a user can access a page they aren't authorized to see just by typing the address, the access control is failing.

5. Security Misconfiguration

Vulnerability Discovered: The server revealed technical details, such as the version of the software being used (e.g., "Apache 2.4.41").

Reasoning: This is a Security Misconfiguration. Keeping default settings or displaying verbose error messages/headers provides attackers with a roadmap of the system's potential weaknesses, making it easier to find specific exploits.

Methodology & Tools Used

To reach these conclusions, the following workflow was applied:

Reconnaissance: Used gobuster to find hidden directories.

Interception: Used Burp Suite to inspect traffic and identify that credentials were being sent without encryption.

Exploitation: Tested input fields for common logic bypasses (SQLi) and verified the lack of rate-limiting by attempting multiple rapid logins.

Question: You discovered that the login page allows an unlimited number of login attempts without trying to slow down the user or lock the account. Whatis the category of this security risk?
- Identification and Authentication Failure.

Question: You noticed that the username and password are sent in cleartext without encryption. What is the category of this security risk?
- Cryptographic Failures

3. Practical Example of Web Application Security

Step-by-Step Execution

Analyze the URL Structure:

After logging into the lab environment, I navigated to my profile page. I observed that the URL contained a numerical parameter:

http://[LAB_IP]/profile?user_id=5

Identify the Vulnerability (IDOR):

I recognized that the user_id was a predictable, sequential integer. This suggested that the application might not be checking if the logged-in user actually has permission to view other IDs.

Perform Parameter Tampering:

I manually clicked into the browser's address bar to edit the URL. I began "fuzzing" or iterating through different ID numbers to see what data the server would return.

I changed user_id=5 to user_id=6 and pressed Enter.

I continued this process, incrementing the value (7, 8, 9...).

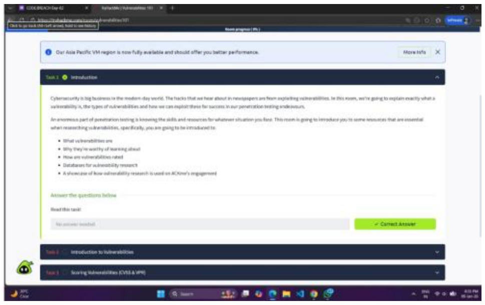
Capture the Flag:

When I changed the parameter to user_id=9, the page refreshed and displayed the profile of Alya (Database Administrator). Because the application did not validate my session against the requested ID, it granted me full access to her "Activity" logs where I see it’s the fake profile so I revert all packages and got the flag.

Result: The hidden flag was revealed within the activity or profile notes for User 10.

Flag Found: THM{IDOR_EXPLORED}

I have attached my screenshots below as proof of completion.



CODEBREACH Day-02
TryHackMe | Vulnerabilities 101
https://tryhackme.com/room/vulnerabilities101
Room progress (9%)

Our Asia Pacific VM region is now fully available and should offer you better performance.

Task 1 Introduction

Cybersecurity is big business in the modern-day world. The hacks that we hear about in newspapers are from exploiting vulnerabilities. In this room, we're going to explain exactly what a vulnerability is, the types of vulnerabilities and how we can exploit these for success in our penetration testing endeavours.

An enormous part of penetration testing is knowing the skills and resources for whatever situation you face. This room is going to introduce you to some resources that are essential when researching vulnerabilities, specifically, you are going to be introduced to:

- What vulnerabilities are
- Why they're worthy of learning about
- How are vulnerabilities rated
- Databases for vulnerability research
- A showcase of how vulnerability research is used on ACKme's engagement

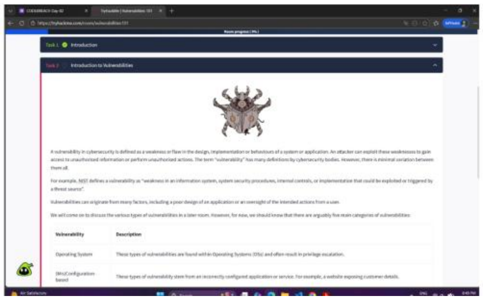
Answer the questions below

Read this task!

No answer needed

Task 2 Introduction to Vulnerabilities

Task 3 Scoring Vulnerabilities (CVSS & VPR)



CODEBREACH Day-02
TryHackMe | Vulnerabilities 101
https://tryhackme.com/room/vulnerabilities101
Room progress (9%)

Task 1 Introduction

Task 2 Introduction to Vulnerabilities

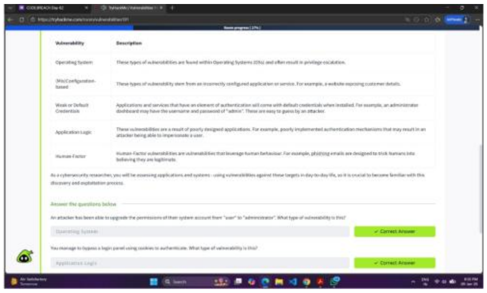
A vulnerability in cybersecurity is defined as a weakness or flaw in the design, implementation or behaviours of a system or application. An attacker can exploit these weaknesses to gain access to unauthorised information or perform unauthorised actions. The term "vulnerability" has many definitions by cybersecurity bodies. However, there is minimal variation between them all.

For example, NIST defines a vulnerability as "weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source".

Vulnerabilities can originate from many factors, including a poor design of an application or an oversight of the intended actions from a user.

We will come on to discuss the various types of vulnerabilities in a later room. However, for now, we should know that there are arguably five main categories of vulnerabilities:

Vulnerability	Description
Operating System	These types of vulnerabilities are found within Operating Systems (OSs) and often result in privilege escalation.
(Mis)Configuration-based	These types of vulnerability stem from an incorrectly configured application or service. For example, a website exposing customer details.



CODEBREACH Day-02
TryHackMe | Vulnerabilities 1
https://tryhackme.com/room/vulnerabilities101
Room progress (27%)

Vulnerability	Description
Operating System	These types of vulnerabilities are found within Operating Systems (OSs) and often result in privilege escalation.
(Mis)Configuration-based	These types of vulnerability stem from an incorrectly configured application or service. For example, a website exposing customer details.
Weak or Default Credentials	Applications and services that have an element of authentication will come with default credentials when installed. For example, an administrator dashboard may have the username and password of "admin". These are easy to guess by an attacker.
Application Logic	These vulnerabilities are a result of poorly designed applications. For example, poorly implemented authentication mechanisms that may result in an attacker being able to impersonate a user.
Human-Factor	Human-Factor vulnerabilities are vulnerabilities that leverage human behaviour. For example, phishing emails are designed to trick humans into believing they are legitimate.

As a cybersecurity researcher, you will be assessing applications and systems - using vulnerabilities against these targets in day-to-day life, so it is crucial to become familiar with this discovery and exploitation process.

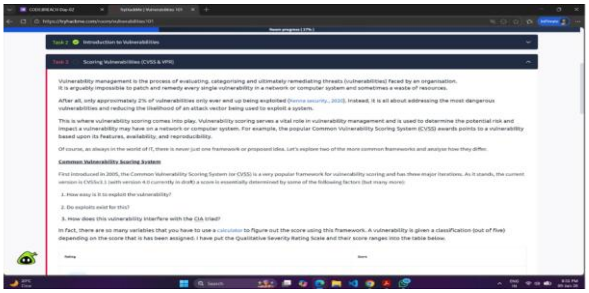
Answer the questions below

An attacker has been able to upgrade the permissions of their system account from "user" to "administrator". What type of vulnerability is this?

Operating System

You manage to bypass a login panel using cookies to authenticate. What type of vulnerability is this?

Application Logic



CODEBREACH Day-02
TryHackMe | Vulnerabilities 101
https://tryhackme.com/room/vulnerabilities101
Room progress (27%)

Task 2 Introduction to Vulnerabilities

Task 3 Scoring Vulnerabilities (CVSS & VPR)

Vulnerability management is the process of evaluating, categorising and ultimately remediating threats (vulnerabilities) faced by an organisation.
It is arguably impossible to patch and remedy every single vulnerability in a network or computer system and sometimes a waste of resources.

After all, only approximately 2% of vulnerabilities only ever end up being exploited (Kenna security., 2020). Instead, it is all about addressing the most dangerous vulnerabilities and reducing the likelihood of an attack vector being used to exploit a system.

This is where vulnerability scoring comes into play. Vulnerability scoring serves a vital role in vulnerability management and is used to determine the potential risk and impact a vulnerability may have on a network or computer system. For example, the popular Common Vulnerability Scoring System (CVSS) awards points to a vulnerability based upon its features, availability, and reproducibility.

Of course, as always in the world of IT, there is never just one framework or proposed idea. Let's explore two of the more common frameworks and analyse how they differ.

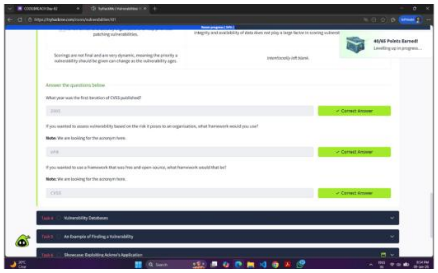
Common Vulnerability Scoring System

First introduced in 2005, the Common Vulnerability Scoring System (or CVSS) is a very popular framework for vulnerability scoring and has three major iterations. As it stands, the current version is CVSSv3.1 (with version 4.0 currently in draft) a score is essentially determined by some of the following factors (but many more):

1. How easy is it to exploit the vulnerability?
2. Do exploits exist for this?
3. How does this vulnerability interfere with the CIA triad?

In fact, there are so many variables that you have to use a calculator to figure out the score using this framework. A vulnerability is given a classification (out of five) depending on the score that is has been assigned. I have put the Qualitative Severity Rating Scale and their score ranges into the table below.

Rating	Score



CODEBREACH Day-02
TryHackMe | Vulnerabilities 1
https://tryhackme.com/room/vulnerabilities101
Room progress (54%)

patching vulnerabilities.	integrity and availability of data does not play a large factor in scoring vulnerabi
Scorings are not final and are very dynamic, meaning the priority a vulnerability should be given can change as the vulnerability ages.	Intentionally left blank.

40/65 Points Earned!
Levelling up in progress...

Answer the questions below

What year was the first iteration of CVSS published?

2005

If you wanted to assess vulnerability based on the risk it poses to an organisation, what framework would you use?

Note: We are looking for the acronym here.

VPR

If you wanted to use a framework that was free and open-source, what framework would that be?

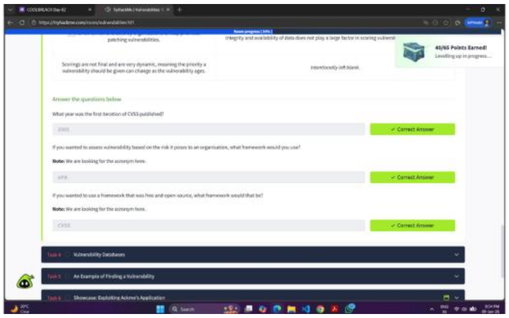
Note: We are looking for the acronym here.

CVSS

Task 4 Vulnerability Databases

Task 5 An Example of Finding a Vulnerability

Task 6 Showcase: Exploiting Ackme's Application



patching vulnerabilities.	integrity and availability of data does not play a large factor in scoring vulnerabi
Scorings are not final and are very dynamic, meaning the priority a vulnerability should be given can change as the vulnerability ages.	Intentionally left blank.

Answer the questions below

What year was the first iteration of CVSS published?

2005

If you wanted to assess vulnerability based on the risk it poses to an organisation, what framework would you use?

Note: We are looking for the acronym here.

VPR

If you wanted to use a framework that was free and open-source, what framework would that be?

Note: We are looking for the acronym here.

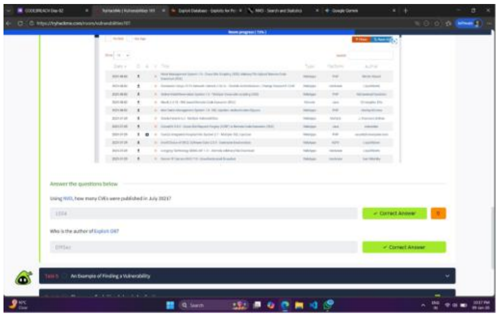
CVSS

Task 4 Vulnerability Databases

Task 5 An Example of Finding a Vulnerability

Task 6 Showcase: Exploiting Ackme's Application

CODEBREACH Day-02
TryHackMe | Vulnerabilities 1
https://tryhackme.com/room/vulnerabilities101
Room progress (54%)
40/65 Points Earned!
Levelling up in progress...
✓ Correct Answer


CODEBREACH Day-02
TryHackMe | Vulnerabilities 101
Exploit Database - Exploits for Pe...
NVD - Search and Statistics
Google Gemini
https://tryhackme.com/room/vulnerabilities101
Room progress (72%)

Answer the questions below

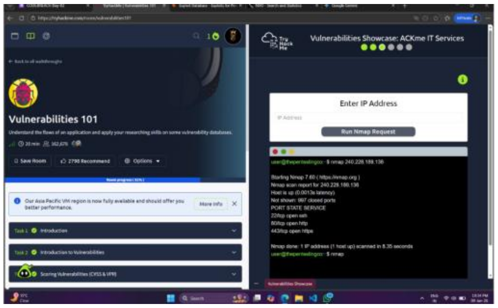
Using NVD, how many CVEs were published in July 2021?

1554 ✓ Correct Answer

Who is the author of Exploit-DB?

OffSec ✓ Correct Answer

Task 5 An Example of Finding a Vulnerability



Vulnerabilities 101

Understand the flaws of an application and apply your researching skills on some vulnerability databases.

20 min 162,676

Our Asia Pacific VM region is now fully available and should offer you better performance.

Task 1 Introduction

Task 2 Introduction to Vulnerabilities

Task 3 Scoring Vulnerabilities (CVSS & VPR)

Vulnerabilities Showcase: ACKme IT Services

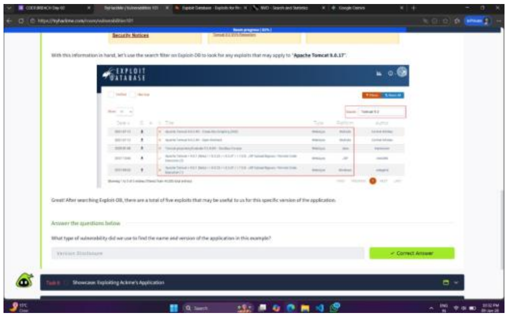
Enter IP Address

IP Address

```
user@thepentestingco:~$ nmap 240.228.188.136

Starting Nmap 7.60 ( https://nmap.org )
Nmap scan report for 240.228.188.136
Host is up (0.0013s latency).
Not shown: 997 closed ports
PORT STATE SERVICE
22/tcp open ssh
80/tcp open http
443/tcp open https

Nmap done: 1 IP address (1 host up) scanned in 8.35 seconds
user@thepentestingco:~$ nmap
```



With this information in hand, let's use the search filter on Exploit-DB to look for any exploits that may apply to "**Apache Tomcat 9.0.17**".

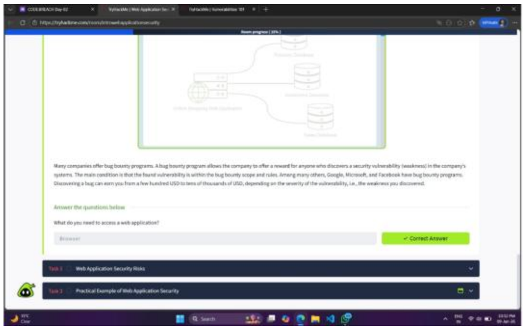
Great! After searching Exploit-DB, there are a total of five exploits that may be useful to us for this specific version of the application.

Answer the questions below

What type of vulnerability did we use to find the name and version of the application in this example?

Version Disclosure ✓ Correct Answer

Task 6 Showcase: Exploiting Ackme's Application


TryHackMe | Web Application Se...
TryHackMe | Vulnerabilities 101
https://tryhackme.com/room/introwebapplicationsecurity
Room progress (25%)
Online Shopping Web Application
Products Database
Customers Database
Sales Database

Many companies offer bug bounty programs. A bug bounty program allows the company to offer a reward for anyone who discovers a security vulnerability (weakness) in the company's systems. The main condition is that the found vulnerability is within the bug bounty scope and rules. Among many others, Google, Microsoft, and Facebook have bug bounty programs. Discovering a bug can earn you from a few hundred USD to tens of thousands of USD, depending on the severity of the vulnerability, i.e., the weakness you discovered.

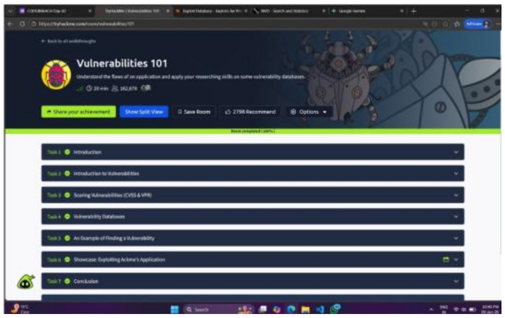
Answer the questions below

What do you need to access a web application?

Browser ✓ Correct Answer

Task 2 Web Application Security Risks

Task 3 Practical Example of Web Application Security



Vulnerabilities 101

Understand the flaws of an application and apply your researching skills on some vulnerability databases.

20 min 162,676

Task 1 Introduction

Task 2 Introduction to Vulnerabilities

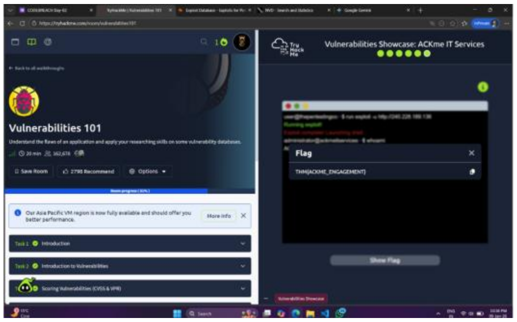
Task 3 Scoring Vulnerabilities (CVSS & VPR)

Task 4 Vulnerability Databases

Task 5 An Example of Finding a Vulnerability

Task 6 Showcase: Exploiting Ackme's Application

Task 7 Conclusion



Vulnerabilities 101

Understand the flaws of an application and apply your researching skills on some vulnerability databases.

Our Asia Pacific VM region is now fully available and should offer you better performance.

Task 1 Introduction

Task 2 Introduction to Vulnerabilities

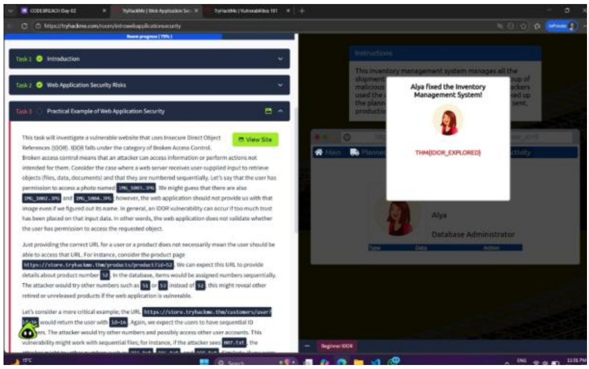
Task 3 Scoring Vulnerabilities (CVSS & VPR)

Vulnerabilities Showcase: ACKme IT Services

```
user@thepentestingco:~$ run exploit -u http://240.228.188.136
Running exploit
Exploit complete. Launching shell
administrator@ackmeitservices:~$ whoami
```

Flag

THM{ACKME_ENGAGEMENT}


CODEBREACH Day-02
TryHackMe | Web Application Se...
TryHackMe | Vulnerabilities 101
https://tryhackme.com/room/introwebapplicationsecurity
Room progress (75%)
View Site
Main
Planned
Beginner IDOR

Task 1 Introduction

Task 2 Web Application Security Risks

Task 3 Practical Example of Web Application Security

This task will investigate a vulnerable website that uses Insecure Direct Object References (IDOR). IDOR falls under the category of Broken Access Control. Broken access control means that an attacker can access information or perform actions not intended for them. Consider the case where a web server receives user-supplied input to retrieve objects (files, data, documents) and that they are numbered sequentially. Let's say that the user has permission to access a photo named `IMG_1001.JPG`. We might guess that there are also `IMG_1002.JPG` and `IMG_1004.JPG`; however, the web application should not provide us with that image even if we figured out its name. In general, an IDOR vulnerability can occur if too much trust has been placed on that input data. In other words, the web application does not validate whether the user has permission to access the requested object.

Just providing the correct URL for a user or a product does not necessarily mean the user should be able to access that URL. For instance, consider the product page `https://store.tryhackme.thm/products/product?id=52`. We can expect this URL to provide details about product number `52`. In the database, items would be assigned numbers sequentially. The attacker would try other numbers such as `51` or `53` instead of `52`; this might reveal other retired or unreleased products if the web application is vulnerable.

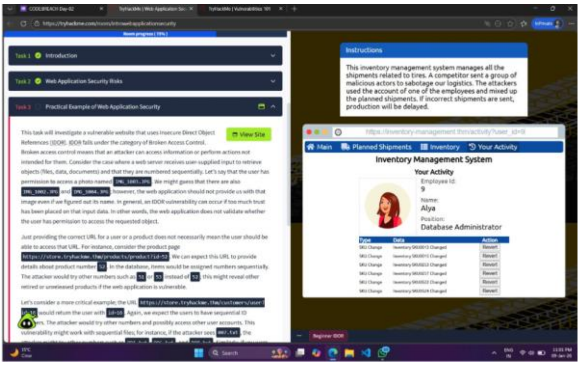
Let's consider a more critical example; the URL `https://store.tryhackme.thm/customers/user?id=16` would return the user with `id=16`. Again, we expect the users to have sequential ID numbers. The attacker would try other numbers and possibly access other user accounts. This vulnerability might work with sequential files; for instance, if the attacker sees `007.txt`, the

Alya fixed the Inventory Management System!

THM{IDOR_EXPLORED}

Alya

Database Administrator


CODEBREACH Day-02
TryHackMe | Web Application Se...
TryHackMe | Vulnerabilities 101
https://tryhackme.com/room/introwebapplicationsecurity
Room progress (75%)
View Site
Main
Planned Shipments
Inventory
Your Activity
Beginner IDOR

Task 1 Introduction

Task 2 Web Application Security Risks

Task 3 Practical Example of Web Application Security

This task will investigate a vulnerable website that uses Insecure Direct Object References (IDOR). IDOR falls under the category of Broken Access Control. Broken access control means that an attacker can access information or perform actions not intended for them. Consider the case where a web server receives user-supplied input to retrieve objects (files, data, documents) and that they are numbered sequentially. Let's say that the user has permission to access a photo named `IMG_1001.JPG`. We might guess that there are also `IMG_1002.JPG` and `IMG_1004.JPG`; however, the web application should not provide us with that image even if we figured out its name. In general, an IDOR vulnerability can occur if too much trust has been placed on that input data. In other words, the web application does not validate whether the user has permission to access the requested object.

Just providing the correct URL for a user or a product does not necessarily mean the user should be able to access that URL. For instance, consider the product page `https://store.tryhackme.thm/products/product?id=52`. We can expect this URL to provide details about product number `52`. In the database, items would be assigned numbers sequentially. The attacker would try other numbers such as `51` or `53` instead of `52`; this might reveal other retired or unreleased products if the web application is vulnerable.

Let's consider a more critical example; the URL `https://store.tryhackme.thm/customers/user?id=16` would return the user with `id=16`. Again, we expect the users to have sequential ID numbers. The attacker would try other numbers and possibly access other user accounts. This vulnerability might work with sequential files; for instance, if the attacker sees `007.txt`, the

Instructions

This inventory management system manages all the shipments related to tires. A competitor sent a group of malicious actors to sabotage our logistics. The attackers used the account of one of the employees and mixed up the planned shipments. If incorrect shipments are sent, production will be delayed.

https://inventory-management.thm/activity?user_id=9

Inventory Management System

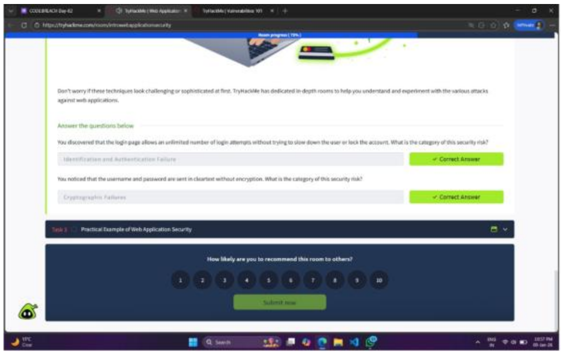
Your Activity

Employee Id: 9

Name: Alya

Position: Database Administrator

Type	Data	Action
SKU Change	Inventory SKU0013 Changed	Revert
SKU Change	Inventory SKU0015 Changed	Revert
SKU Change	Inventory SKU0012 Changed	Revert
SKU Change	Inventory SKU0017 Changed	Revert
SKU Change	Inventory SKU0022 Changed	Revert
SKU Change	Inventory SKU0024 Changed	Revert


CODEBREACH Day-02
TryHackMe | Web Applicatio...
TryHackMe | Vulnerabilities 101
https://tryhackme.com/room/introwebapplicationsecurity
Room progress (75%)
Submit now

Don't worry if these techniques look challenging or sophisticated at first. TryHackMe has dedicated in-depth rooms to help you understand and experiment with the various attacks against web applications.

Answer the questions below

You discovered that the login page allows an unlimited number of login attempts without trying to slow down the user or lock the account. What is the category of this security risk?

Identification and Authentication Failure ✓ Correct Answer

You noticed that the username and password are sent in cleartext without encryption. What is the category of this security risk?

Cryptographic Failures ✓ Correct Answer

Task 3 Practical Example of Web Application Security

How likely are you to recommend this room to others?

1 2 3 4 5 6 7 8 9 10