



Sri Lanka Institute of Information Technology

Final Project Report

ISP Project Report

Information Security Project 2021

Project ID: **ISP-21-REG-02**

Submitted by:

| IT Number | Name |
|------------|-------------------------|
| IT19056012 | Meeriyagalla P.Y. |
| IT19147192 | K.A.P.P. Wickramarathne |

Date of submission: 14/11/2021

Abstract

Transportation systems are critical infrastructures in every country. IoT and cloud computing systems are rapidly growing in today's world and those systems are also used to provide some services on transportation sector. The main purpose to create this CTF box is to elevate the need of having a good security system in cloud base transportation systems. This CTF box completely developed on AWS (Amazon Web Service) cloud platform. By playing this CTF box the professionals in the industry and the developers who develop cloud base web systems, can gain knowledge about security vulnerabilities and misconfigurations of website development and cloud platform configurations.

Acknowledgement

We would like to express our gratitude for the Sri Lanka Institute of Information Technology for giving this opportunity to get professional experiences about a development and implementing project of a CTF (Capture the Flag) Box. We would like to convey our profound thanks to Information Security Project (ISP) lecture in charge Dr.Lakmal Rupasinghe , and the subject instructor Miss. Menaka Moonamaldeniya, Miss. Chathuri Udagedara for their excitement, patience, insightful remarks, valuable information, practical guidance, and never-ending ideas, all of which have greatly aided us during our CTF creating project.

We are grateful for every individual and organization who helped us throughout the creation of the CTF box by providing cloud services (AWS), domain services (Domain.com), and tutorials (Mr. Neal Davis - YouTube).

Declaration

We declare that this project report or part of it was not a copy of a document done by any organization, university any other institute or a previous student project group at SLIIT and was not copied from the Internet or other sources.

Project Details

| | |
|---------------|----------------------|
| Project Title | Trip-To-Hell CTF Box |
| Project ID | ISP-21-REG-02 |

Group Members

| Reg. No | Name | Signature |
|------------|-------------------------|---|
| IT19056012 | Meeriyagalla P.Y. |  |
| IT19147192 | K.A.P.P. Wickramarathne |  |

Table of Contents

| | |
|--|------------|
| Abstract..... | i |
| Acknowledgement..... | iii |
| Declaration..... | iv |
| Table of Contents | v |
| List of Figures..... | vi |
| List of Tables | vii |
| 1. Introduction..... | 1 |
| 1.1 Problem Statement..... | 1 |
| 1.2 Product Scope..... | 1 |
| 1.3 Project Report Structure | 2 |
| 2. Methodology | 3 |
| 2.1 Requirements and Analysis | 3 |
| 2.2 Design..... | 3 |
| 2.3 Implementation..... | 7 |
| 2.4 Testing | 21 |
| 3. Evaluation | 24 |
| 3.1 Assessment of the Project results | 24 |
| 3.2 Lessons Learned | 25 |
| 3.3 Future Work..... | 26 |
| 4. Conclusion | 27 |
| 5. References | 28 |
| Appendix A: Test Results..... | 6 |

List of Figures

1. Figure 01-CTF flow chart: 4 (Page number)
2. Figure 02-TravelMate Deployment: 5
3. Figure 03-Reservation Deployment: 5
4. Figure 04-Task 09 Deployment: 6
5. Figure 05-ER Diagram: 6
6. Figure 06-WordPress Backend: 7
7. Figure 07-I Am User Dashboard : 7
8. Figure 08-I Am user : 8
9. Figure 09-VPC : 8
10. Figure 10- VPC : 9
11. Figure 11-Security Rules : 9
12. Figure 12-Security Rules : 9
13. Figure 13-Security Policies : 10
14. Figure 14-EC 2 Instance Creation : 10
15. Figure 15-EC 2 Instance Creation : 11
16. Figure 16-EC 2 Instance Creation : 11
17. Figure 17-EC 2 Instance Creation : 12
18. Figure 18-EC 2 Instance Creation : 12
19. Figure 19-EC 2 RDS : 13
20. Figure 20-EC 2 Instance N.Virginia : 13
21. Figure 21-EC 2 Instance Singapore : 14
22. Figure 22-Wordpress Connection : 14
23. Figure 23- Wordpress Backend : 15
24. Figure 24-Wordpress Install Code : 15
25. Figure 25- Connecting To RDS DB : 16
26. Figure 26- Connecting To RDS DB : 16
27. Figure 27- Connecting To RDS DB : 17
28. Figure 28-Route 53 : 17
29. Figure 29-Route 53 : 18

- 30. Figure 30-Route Plugins : 18
- 31. Figure 31-Route Plugins : 19
- 32. Figure 32-S3 Bucket : 19
- 33. Figure 33-TryHackMe : 20
- 34. Figure 34-TryHackMe : 20
- 35. Figure 35-PenTest ZAP : 21
- 36. Figure 36-PenTest Nikto : 22
- 37. Figure 37-PenTest Wapiti : 22
- 38. Figure 38-PenTest Skipfish : 23

List of Tables

List of Acronyms and Abbreviations

IoT: Internet of Things

AWS: Amazon Web Service (Cloud Service Providing Platform)

CTF: Capture the Flag (Game that can play to improve cyber security skills)

RDS: Relational Database Service

VPC: Virtual Private Cloud

1. Introduction

1.1 Problem Statement

In most of industries cloud computing is more popular today. Transportation system is a critical infrastructure in every country. Most of the countries in the world use IoT technologies and web base systems with cloud computing in nowadays to improve the comfort and with the purpose of time saving. When a transportation system improvises with IoT, cloud base systems, and web technologies it can lead to critical cyber security threats. Even a smaller threat can be cause serious consequences in transportation sector. The main intention of creating this CTF box is to give a hand on experience of the security threats that can occur with above mentioned technologies.

To develop this Trip-To-Hell CTF box mainly used platform is AWS. For this CTF project a whole fully functioning website (www.travelmatetth.com) was developed using WordPress. The main reason to use WordPress for the development of the website is because, it is commonly used by professional developers to create and design websites. To get the real-life experiences about these services all the tools and services that used to build this project are up to date. There can be critical information security threats even the systems are up to date. That is the main reason to use latest versions of services and systems. To improve the realism, a domain name was purchased from Domain.com.

There is main 09 levels in this Trip-To-Hell CTF box. Each level has sublevels. This CTF box is fully built in AWS cloud platform. Because of that there is no image file that need to be installed by the players when playing this CTF. Players can easily open the website (www.travelmatetth.com) and play the CTF by joining with www.tryhackme.com (CTF url: tryhackme.com/jr/triptohellctf).

1.2 Product Scope

In Trip-to-Hell CTF box, the players can gather knowledge about new systems which contains vulnerabilities on transportation schedule and reservation management system. There will be

cloud base servers, web-based systems, and vulnerabilities that players can exploit and gather information which is related to railway scheduling and reservation system. By this project, people who are related to railway management systems and information security professionals will be able to identify real-life types of attacks with the knowledge that gain through the CTF box.

1.3 Project Report Structure

The implementation of the CTF box back-end and the website creating details will be explain in the methodology sector separately. Details about the levels, achievements and future development areas explained in the evaluation sector.

2. Methodology

2.1 Requirements and Analysis

The vulnerable website contains railway management details including railway path details, train scheduling details and it only shows the details about train seats reservations. The player can investigate the railway scheduling details and seat reservation details. Players need to find vulnerable loopholes to break into the website, scheduling system and to connect to the hidden servers.

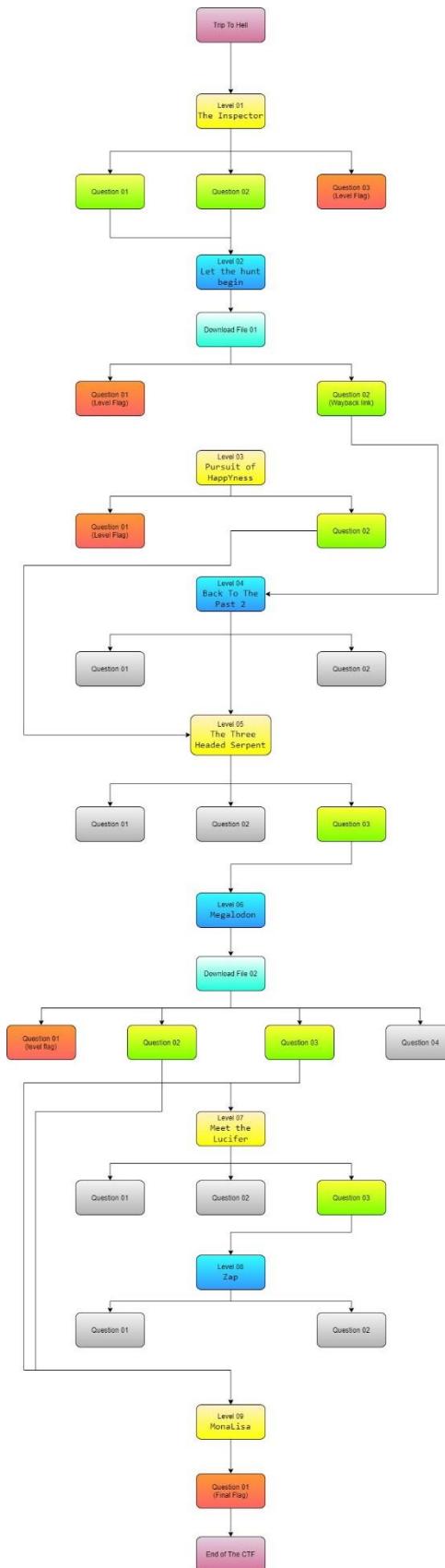
As the CTF player he or she will not need to download an image file and host it locally. The complete CTF is developed on AWS cloud. This CTF can play with both Linux and Windows operating systems (Linux OS is highly recommended). To play this CTF box player need to have security tools. According to the tasks that must play in this CTF the recommended tools:

- Wireshark
- Nmap
- OWASP ZAP
- Deepsound
- CryptTool
- STool
- Web Browser
- Winrar
- PuTTY (for Windows users)
- Nikto or any scanner
- Hydra

Player also need to register to the Try Hack Me website.

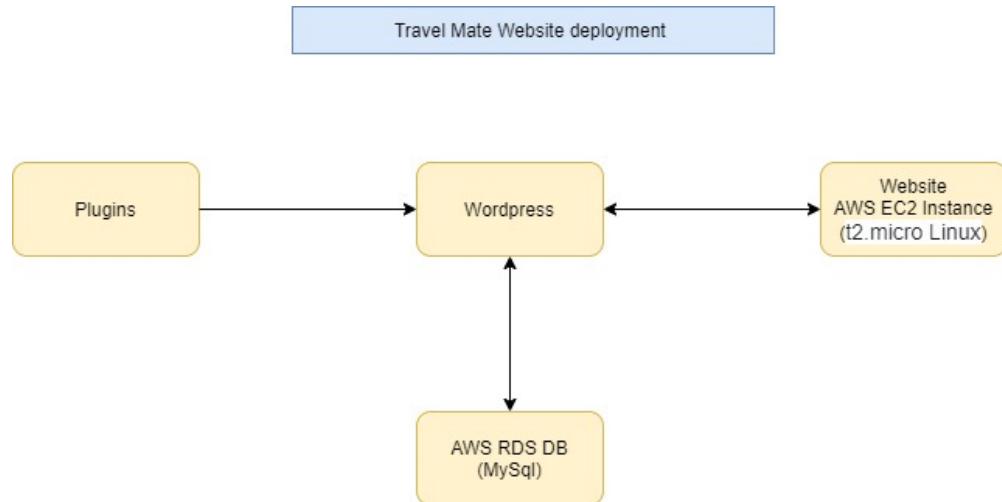
2.2 Design

The workflow of the CTF box is as below.



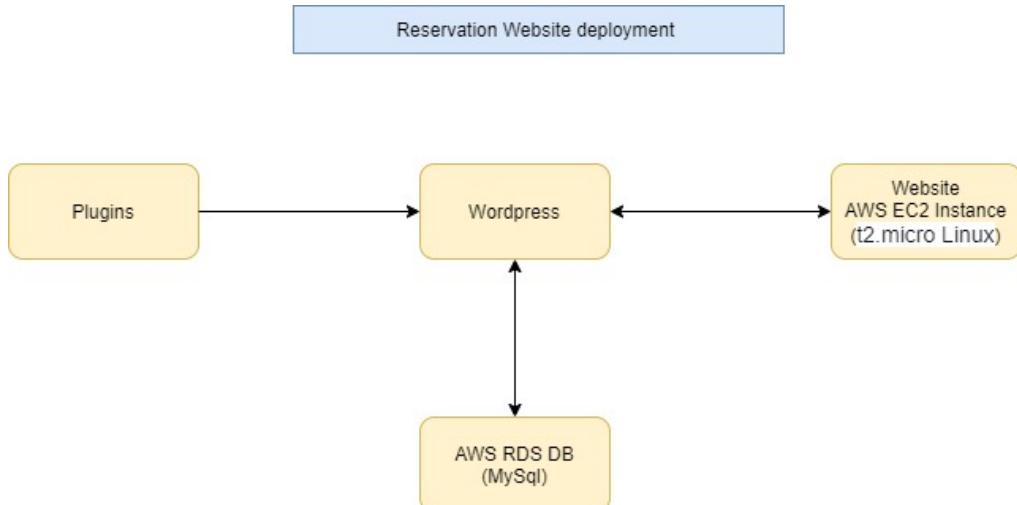
[Figure 01-CTF flow chart]

The basic deployment chart of the Travel Mate (www.travellmateth.com) website.



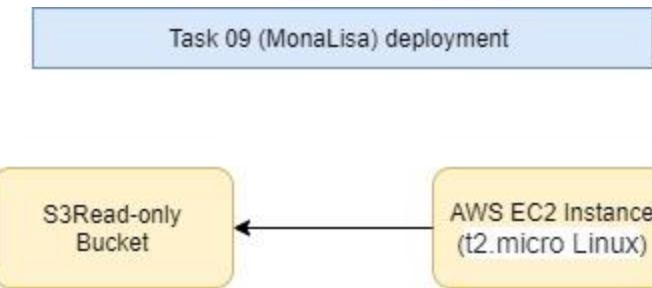
[Figure 02-TravelMate Deployment]

Reservation page deployment.



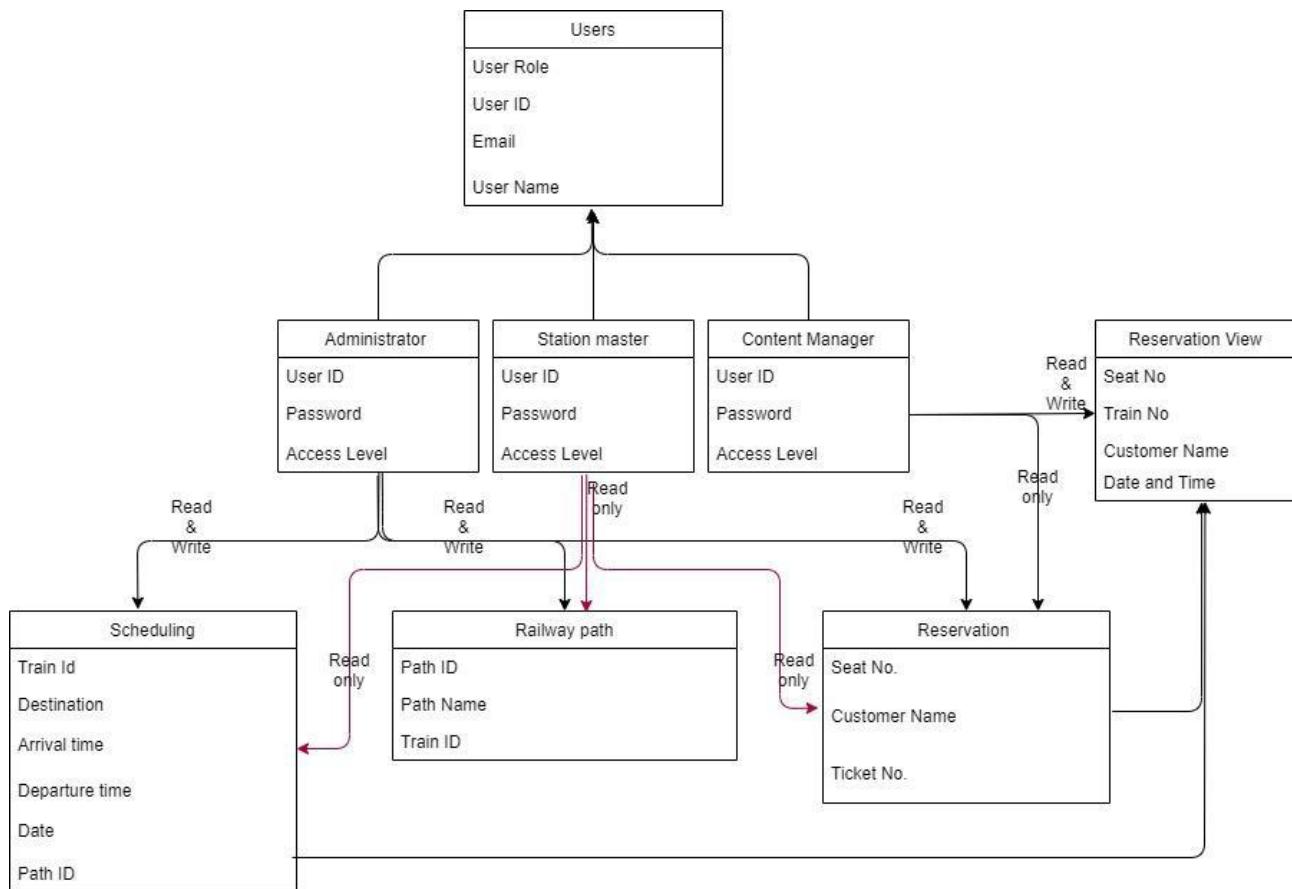
[Figure 03-Reservation Deployment]

Task 09 deployment.



[Figure 04-Task 09 Deployment]

The ER-Diagram for the database.



[Figure 05-ER Diagram]

Administrator: Administrator

Station master: Contributor

Content Manager: Editor

| Username | Name | Email | Role | Posts | Status |
|--------------|---------------------|---------------------------|---------------|-------|----------|
| Hunter | Will Hunter | fblevel03pass@gmail.com | Agent | 0 | Approved |
| Yashashmi | Anjela Fernando | it19056012@my.slit.lk | Contributor | 0 | Approved |
| Oliver | Oliver Stephen | OliverStephen@gmail.com | Editor | 0 | Approved |
| Olivia | Olivia Macdovel | OliviaMacdovel@gmail.com | Subscriber | 0 | Approved |
| Kevin | Levin Fernando | KeinLevin@gmail.com | Subscriber | 0 | Approved |
| Wonder_Women | Pamudi Meeriyagalla | pamudiyashashmi@gmail.com | Subscriber | 0 | Approved |
| Will Seeker | — | travelmateth@gmail.com | Administrator | 1 | Approved |

[Figure 06-WordPress Backend]

2.3 Implementation

The whole project was developed on the AWS cloud platform because when creating the project, the mainly focused areas were cloud computing and web security. As the starting point of the project, created an AWS account. After that created a user role as I am user.

| User groups | Users | Roles | Policies | Identity providers |
|-------------|-------|-------|----------|--------------------|
| 1 | 1 | 10 | 3 | 0 |

[Figure 07-I Am User Dashboard]

The screenshot shows the AWS Identity and Access Management (IAM) service interface. On the left, there's a navigation sidebar with options like 'Identity and Access Management (IAM)', 'Access management', 'Access reports', and 'Roles'. The 'Roles' section is currently selected. The main content area is titled 'Roles (10) Info' and contains a table listing ten roles. The columns in the table are 'Role name', 'Trusted entities', and 'Last activity'. The roles listed are: Admin_access, AWSServiceRoleForGlobalAccelerator, AWSServiceRoleForRDS, AWSServiceRoleForSupport, AWSServiceRoleForTrustedAdvisor, EC2_Read, New_Role, S3-All-Access, S3-Full-Access, and S3-Read-Only. Each role entry includes the service it's associated with and the last time it was used.

[Figure 08-I Am user]

Then created a VPC (Virtual Private Cloud) on AWS platform.

The screenshot shows the AWS Virtual Private Cloud (VPC) service interface. On the left, there's a navigation sidebar with options like 'VPC Dashboard', 'EC2 Global View', 'Your VPCs', 'Subnets', 'Route Tables', 'Internet Gateways', 'Egress Only Internet Gateways', 'Carrier Gateways', 'DHCP Options Sets', 'Elastic IPs', 'Managed Prefix Lists', 'Endpoints', 'Endpoint Services', 'NAT Gateways', and 'Peering'. The 'Your VPCs' section is currently selected. The main content area is titled 'Your VPCs (1/1) Info' and contains a table with one row. The columns in the table are 'Name', 'VPC ID', 'State', 'IPv4 CIDR', and 'IPv6 CIDR (Network border group)'. The single VPC entry is named 'vpc-31c6bc4c', has a VPC ID of 'vpc-31c6bc4c', is in 'Available' state, has an IPv4 CIDR of '172.31.0.0/16', and no IPv6 CIDR or Network border group assigned. Below the table, there's a 'Details' section with more VPC configuration details.

[Figure 09-VPC]

The screenshot shows the AWS VPC Subnets page. On the left, there's a sidebar with links for VPC Dashboard, EC2 Global View, Filter by VPC, VIRTUAL PRIVATE CLOUD, Your VPCs, Subnets (selected), Route Tables, Internet Gateways, Egress Only Internet Gateways, Carrier Gateways, DHCP Options Sets, Elastic IPs, Managed Prefix Lists, Endpoints, Endpoint Services, NAT Gateways, and Peering. The main area displays a table titled "Subnets (6) Info" with columns: Name, Subnet ID, State, VPC, IPv4 CIDR, and IPv6 CIDR. The subnets listed are: subnet-fb8cf1ca, subnet-122dce13, subnet-592cd515, subnet-78e1c81e, subnet-229ca803, and subnet-6e032931, all in an "Available" state. Below the table, there's a section titled "Select a subnet" with a large input field containing "[Figure 10- VPC]".

Then created security rules (Rule name: web-access) for the VPC using security group.

The screenshot shows the AWS Security Groups page. The sidebar includes links for Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, and Images. The main area shows a table titled "Security Groups (2) Info" with columns: Name, Security group ID, Security group name, VPC ID, Description, and Owner. It lists two groups: "sg-0afbe508ad55a596d" named "Web-Access" and "sg-681fe979" named "default". Both are associated with the VPC "vpc-31c6bc4c". Below the table, there's a section titled "Select a security group" with a large input field containing "[Figure 11-Security Rules]".

The screenshot shows the AWS Security Group details page for the "Web-Access" security group. The sidebar includes links for Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images, AMIs, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, and Network Interfaces. The main area shows summary statistics: Owner (916973023958), Inbound rules count (5 Permission entries), and Outbound rules count (4 Permission entries). Below this, there are tabs for "Inbound rules" (selected), "Outbound rules", and "Tags". A note says "You can now check network connectivity with Reachability Analyzer" with a "Run Reachability Analyzer" button. The "Inbound rules (5)" table has columns: Name, Security group rule..., IP version, Type, Protocol, and Port range. The rules listed are:

| Name | Security group rule... | IP version | Type | Protocol | Port range |
|------|------------------------|------------|--------------|----------|------------|
| - | sgr-02682912eee5061... | IPv4 | SSH | TCP | 22 |
| - | sgr-0e1766d86bba47... | IPv4 | MySQL/Aurora | TCP | 3306 |
| - | sgr-02568502241828... | IPv4 | All TCP | TCP | 0 - 65535 |
| - | sgr-067c360b3e7009ece | IPv4 | HTTP | TCP | 80 |
| - | sgr-06bf43925ffe0ae8f | IPv4 | All traffic | All | All |

Below the table, there are buttons for "Manage tags" and "Edit inbound rules". At the bottom, there's a large input field containing "[Figure 12-Security Rules]".

To add more security for the VPC created security policies.

The screenshot shows the AWS Identity and Access Management (IAM) Policies page. The left sidebar includes sections for Identity providers, Account settings, Access management (User groups, Users, Roles), Policies (selected), Access reports (Access analyzer, Archive rules, Analyzers, Settings, Credential report, Organization activity, Service control policies (SCPs)), and Feedback. The main content area displays a table titled 'Policies (882) Info' with columns for Policy name, Type, Used as, and Description. Policies listed include 'Created_EC2', 'S3-All-Access', 's3_policy', 'AWSDirectConnectReadOnlyAccess', 'AmazonGlacierReadOnlyAccess', 'AWSMarketplaceFullAccess', 'ClientVPNServiceRolePolicy', 'AWSSSOAdministrator', 'AWSIoT1ClickReadOnlyAccess', 'AutoScalingConsoleReadOnlyAccess', and 'AmazonDMSRedshiftS3Role'. A search bar at the top allows filtering by property or policy name. A 'Create Policy' button is located in the top right corner. The bottom of the page includes links for Feedback, English (US), Amazon Web Services, Inc. or its affiliates, All rights reserved., Privacy Policy, Terms of Use, and Cookie preferences.

[Figure 13-Security Policies]

To create the main website (www.travellmateth.com) and the reservation website created a t2.micro EC2 Linux instance on AWS. To make the IP address a static one, added an elastic IP to the EC2 instance.

The screenshot shows the AWS EC2 Instance Creation wizard, Step 2: Choose an Instance Type. The top navigation bar includes links for Choose AMI, Choose Instance Type, Configure Instance, Add Storage, Add Tags, Configure Security Group, and Review. Below the navigation, a note states: 'Step 2: Choose an Instance Type' and 'Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.' A filter section allows filtering by instance families (All instance families, Current generation) and show/hide columns. A note says 'Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, ~ 1 GiB memory, EBS only)'. The main table lists t2 instance types: t2.nano, t2.micro (Free tier eligible), t2.small, t2.medium, t2.large, t2.xlarge, and t2.2xlarge. Columns include Family, Type, vCPUs, Memory (GiB), Instance Storage (GB), EBS-Optimized Available, Network Performance, and IPv6 Support. Buttons at the bottom include Cancel, Previous, Review and Launch (highlighted in blue), and Next: Configure Instance Details. The bottom of the page includes links for Feedback, English (US), Amazon Web Services, Inc. or its affiliates, All rights reserved., Privacy Policy, Terms of Use, and Cookie preferences.

[Figure 14-EC 2 Instance Creation]

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.

A copy of a tag can be applied to volumes, instances or both.

Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

| Key | (128 characters maximum) | Value | (256 characters maximum) | Instances | Volumes | Network Interfaces |
|-----|--------------------------|-------------------------------------|-------------------------------------|-------------------------------------|---------|--------------------|
| ith | on-test | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | |

Add another tag (Up to 50 tags maximum)

[Figure 15-EC 2 Instance Creation]

Cancel Previous Review and Launch Next: Configure Security Group

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

⚠ Improve your instances' security. Your security group, Web-Access, is open to the world.

Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only.

You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

AMI Details [Edit AMI](#)

| |
|---|
| Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-082105f875acab993 |
| Free tier eligible |
| Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is a... |
| Root Device Type: ebs Virtualization type: hvm |

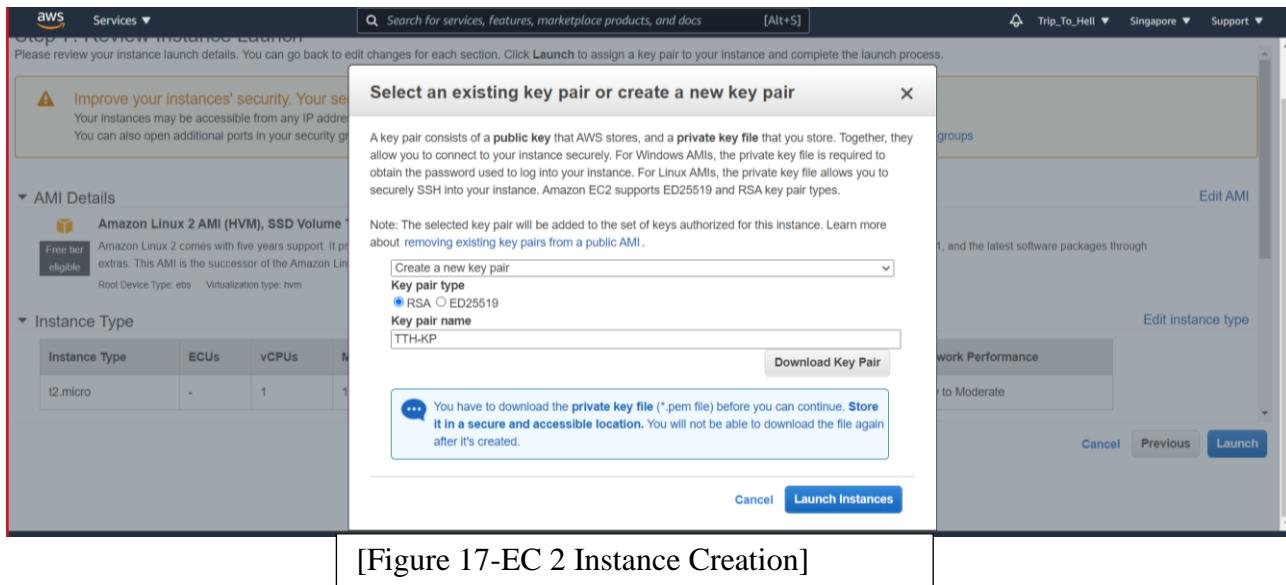
Instance Type [Edit instance type](#)

| Instance Type | ECUs | vCPUs | Memory (GiB) | Instance Storage (GB) | EBS-Optimized Available | Network Performance |
|---------------|------|-------|--------------|-----------------------|-------------------------|---------------------|
| t2.micro | - | 1 | 1 | EBS only | - | Low to Moderate |

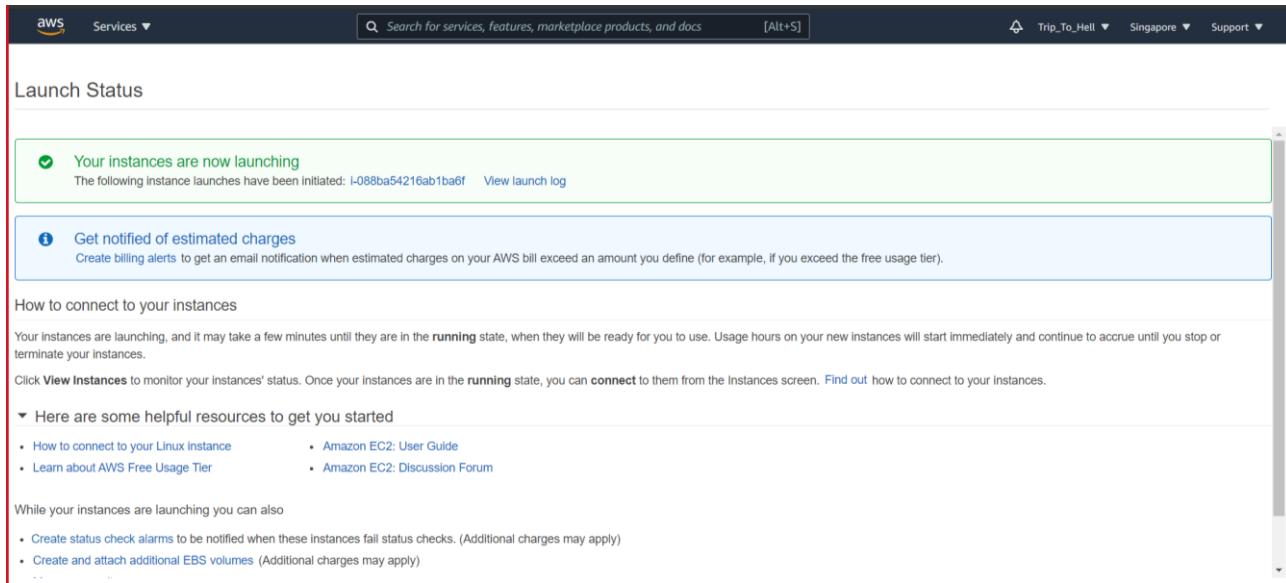
Security Groups [Edit security groups](#)

[Figure 16-EC 2 Instance Creation]

Cancel Previous **Launch**



[Figure 17-EC 2 Instance Creation]



[Figure 18-EC 2 Instance Creation]

For the database system created a RDS (Relational Database Service) MySQL database on AWS.

| DB identifier | Role | Engine | Region & AZ | Size | Status | CPU |
|---------------|----------|-----------------|-------------|-------------|-----------|-----|
| mysqldbth | Instance | MySQL Community | us-east-1f | db.t2.micro | Stopped | - |
| travelmate | Instance | MySQL Community | us-east-1b | db.t2.micro | Available | 1 |

Selected region for the Travel Mate website implementation is North Virginia. For the reservation website page selected Singapore region.

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPv4 DNS |
|-------------------|---------------------|----------------|---------------|-------------------|--------------|-------------------|-----------------|
| Travel Mate wi... | i-09621c2f69b96ee9f | Running | t2.micro | 2/2 checks passed | No alarms | us-east-1b | ec2-44-199-17-1 |
| victorthevillain | i-010db61480e9681cd | Stopped | t2.micro | - | No alarms | us-east-1b | ec2-44-199-87-1 |
| New Travel Ma... | i-01dc78657af3b7dce | Stopped | t2.micro | - | No alarms | us-east-1c | ec2-52-21-132-1 |

The screenshot shows the AWS EC2 Instances page. The left sidebar includes options like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances (selected), Images, AMIs, and Elastic Block Store. The main content area displays a table of instances:

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPv4 DNS |
|---------------------|---------------------|----------------|---------------|-------------------|--------------|-------------------|-----------------|
| 2nd | i-013f3e3b2c195ab39 | Stopped | t2.micro | - | No alarms | ap-southeast-1b | - |
| ttctf | i-024040770bc3aaaf0 | Running | t2.micro | 2/2 checks passed | No alarms | ap-southeast-1b | ec2-13-229-178- |
| sql inject singa... | i-0eadae238920a44d | Running | t2.micro | 2/2 checks passed | No alarms | ap-southeast-1c | ec2-52-74-61-15 |

A modal window titled "Select an instance above" is open at the bottom of the table.

[Figure 21-EC 2 Instance Singapore]

To connect the database and the instance used WordPress.

The screenshot shows the WordPress database connection setup page. It features a large blue "W" logo at the top. Below it, a form asks for database connection details:

Below you should enter your database connection details. If you're not sure about these, contact your host.

| | | |
|---------------|--|--|
| Database Name | <input type="text" value="travelmate"/> | The name of the database you want to use with WordPress. |
| Username | <input type="text" value="JhonWickAD"/> | Your database username. |
| Password | <input type="text" value="d4mnth155h1t"/> | Your database password. |
| Database Host | <input type="text" value="travelmate.c61uczwdf1g.us-eas"/> | You should be able to get this info from your web host, if localhost doesn't work. |
| Table Prefix | <input type="text" value="wp_"/> | If you want to run multiple WordPress installations in a single database, change this. |

[Figure 22-Wordpress Connection]

The screenshot shows the WordPress Backend User Management page. The left sidebar includes links for Dashboard, Posts, Media, Pages, Comments (7), TablePress, Ultimate Member, Elementor, Templates, WP Popups, Appearance, Plugins (2), and Users. The main area displays a table of users:

| Username | Name | Email | Role | Posts | Status |
|--------------|---------------------|---------------------------|---------------|-------|----------|
| Hunter | Will Hunter | fbfilevel03pass@gmail.com | Agent | 0 | Approved |
| Yashashmi | Anjela Fernando | it19056012@my.slit.lk | Contributor | 0 | Approved |
| Oliver | Oliver Stephen | OliverStephen@gmail.com | Editor | 0 | Approved |
| Olivia | Olivia Macdovel | OliviaMacdovel@gmail.com | Subscriber | 0 | Approved |
| Kevin | Levin Fernando | KeinLevin@gmail.com | Subscriber | 0 | Approved |
| Wonder_Women | Pamudi Meeriyagalla | pamudiyashashmi@gmail.com | Subscriber | 0 | Approved |
| Will Seeker | — | travelmateth@gmail.com | Administrator | 1 | Approved |
| Username | Name | Email | Role | Posts | Status |

[Figure 23- Wordpress Backend]

When connecting the databases there were configuration commands that needed to run on the Linux instance.

```
# Install WordPress on EC2 using RDS MySQL DB
# Either add "sudo" before all commands or use "sudo su" first
yum update -y
amazon-linux-extras install -y php7.2
yum install -y httpd
systemctl start httpd
systemctl enable httpd
cd /var/www/html
wget https://wordpress.org/latest.tar.gz
tar -xzf latest.tar.gz
cp -r wordpress/* ./
chmod -R 755 wp-content
```

To do that, connected to the Linux instance using security key pair with Kali Linux terminal.

Connect to the database using linux:

[Figure 24-Wordpress Install Code]

```

File Machine View Input Devices Help
Mozilla Firefox  ~/Desktop/ctf.txt - Mous... OWASP ZAP - OWASP ... terminal 202
root@ip-172-31-84-231:/home/ec2-user
File Actions Edit View Help
[~(kali㉿kali)-~/Desktop]
└─$ sudo su
[sudo] password for kali:
[~(root㉿kali)-~/home/kali/Desktop]
└─# chmod 400 TTH-KP2V.pem
[~(root㉿kali)-~/home/kali/Desktop]
└─# ssh -i "TTH-KP2V.pem" ec2-user@44.199.17.110
Last login: Fri Nov 12 18:50:38 2021 from 116.206.246.253
|_| ( _ _ ) Amazon Linux 2 AMI
https://aws.amazon.com/amazon-linux-2/
12 package(s) needed for security, out of 28 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-84-231 ~]$ sudo su
[root@ip-172-31-84-231 ~]# mysql -h travelmate.c61uczwdfr3.us-east-1.rds.amazonaws.com -P 3306 -u JhonWickAD -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 23553
Server version: 8.0.23 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]>

```

[Figure 25- Connecting To RDS DB]

```

File Machine View Input Devices Help
Mozilla Firefox  ~/Desktop/ctf.txt - Mous... OWASP ZAP - OWASP ... terminal 202
root@ip-172-31-84-231:/home/ec2-user
File Actions Edit View Help
MySQL [(none)]> show database;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'database' at line 1
MySQL [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| travelmate |
+-----+
5 rows in set (0.03 sec)

MySQL [(none)]> use travelmate;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [travelmate]> show tables;
+-----+
| Tables_in_travelmate |
+-----+
| wp_actionscheduler_actions |
| wp_actionscheduler_claims |
| wp_actionscheduler_groups |
| wp_actionscheduler_logs |
| wp_aioseo_cache |
| wp_aioseo_notifications |
| wp_aioseo_posts |
| wp_commentmeta |
| wp_comments |
| wp_easy_query |
| wp_frm_fields |
| wp_frm_forms |
| wp_frm_item_metas |
| wp_frm_items |
| wp_links |
| wp_ninja_table_items |
| wp_options |
| wp_postmeta |
| wp_posts |
+-----+

```

[Figure 26- Connecting To RDS DB]

```

File Machine View Input Devices Help
Mozilla Firefox
OWASP ZAP - OWASP...
root@ip-172-31-84-231:~#
12:31 PM
File Actions Edit View Help
5 rows in set (0.03 sec)

MySQL [(none)]> use travelmate;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [travelmate]> show tables;
+-----+
| Tables_in_travelmate |
+-----+
| wp_actionscheduler_actions |
| wp_actionscheduler_claims |
| wp_actionscheduler_groups |
| wp_actionscheduler_logs |
| wp_aioseo_cache |
| wp_aioseo_notifications |
| wp_aioseo_posts |
| wp_commentmeta |
| wp_comments |
| wp_easy_query |
| wp_frm_fields |
| wp_frm_forms |
| wp_frm_item_metas |
| wp_frm_items |
| wp_links |
| wp_ninja_table_items |
| wp_options |
| wp_postmeta |
| wp_posts |
| wp_relationships |
| wp_term_taxonomy |
| wp_termmeta |
| wp_terms |
| wp_test1 |
| wp_um_metadata |
| wp_usermeta |
| wp_users |
+-----+
27 rows in set (0.00 sec)

MySQL [travelmate]>

```

[Figure 27- Connecting To RDS DB]

To make the CTF project more realistic, purchased a domain from Domain.com and added it using route 53 feature.

The screenshot shows the AWS Route 53 dashboard. On the left, there's a sidebar with navigation links like Dashboard, Hosted zones, Health checks, Traffic flow, Domains, Resolver, and DNS Firewall. The main area has a blue header bar with a message about the new Route 53 console. Below the header, there's a summary section with four cards: 'DNS management' (1 Hosted zone), 'Traffic management' (Create policy button), 'Availability monitoring' (0 checks, Create health check button), and 'Domain registration' (Register domain button). At the bottom, there are links for Feedback, English (US), and various AWS terms like Privacy Policy, Terms of Use, and Cookie preferences.

[Figure 28-Route 53]

Records (4) Info
Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.

| Record name | Type | Routing... | Differ... | Value/Route traffic to |
|------------------------|-------|------------|-----------|--|
| travelmatetth.com | A | Simple | - | 44.199.17.110 |
| travelmatetth.com | NS | Simple | - | ns-1230.awsdns-25.org. ns-1686.awsdns-18.co.uk. ns-333.awsdns-41.com. ns-678.awsdns-20.net. |
| travelmatetth.com | SOA | Simple | - | ns-1230.awsdns-25.org. awsdns-hostmaster.amazon.com. |
| www.travelmatetth.c... | CNAME | Simple | - | travelmatetth.com |

[Figure 29-Route 53]

For develop and design purposes of the Travel Mate website used WordPress and plugins. Not only for the Travel Mate website but also for the reservation web page used WordPress and plugins for development purposes.

| Plugin | Description | Automatic Updates |
|-------------------|--|---------------------|
| Akismet Anti-Spam | Used by millions, Akismet is quite possibly the best way in the world to protect your blog from spam. It keeps your site protected even while you sleep. To get started: activate the Akismet plugin and then go to your Akismet Settings page to set up your API key. Version 4.1.12 By Automatic View details | Enable auto-updates |
| All in One SEO | SEO for WordPress. Features like XML Sitemaps, SEO for custom post types, SEO for blogs, business sites, ecommerce sites, and much more. More than 80 million downloads since 2007. Version 4.1.5.1 By All in One SEO Team View details Suggest a Feature | Enable auto-updates |
| Easy Query | A query builder plugin for WordPress. Version 2.0.4 By Darren Cooney View details | Enable auto-updates |
| Elementor | The Elementor Website Builder has it all: drag and drop page builder, pixel perfect design, mobile responsive editing, and more. Get started now! Version 3.4.4 By Elementor.com View details Docs & FAQs Video Tutorials | Enable auto-updates |
| Formidable Forms | Quickly and easily create drag-and-drop forms Version 5.0.12 By Strategy11 View details | Enable auto-updates |

[Figure 30-Route Plugins]

The screenshot shows the WordPress admin dashboard under the 'Plugins' section. On the left sidebar, 'Plugins' is selected, showing 2 installed and 1 available. The main area lists several plugins:

- Formidable Forms**: Quickly and easily create drag-and-drop forms. Version 5.0.12 | By Strategy11 | View details. Enable auto-updates.
- Hello Dolly**: This is not just a plugin, it symbolizes the hope and enthusiasm of an entire generation summed up in two words sung most famously by Louis Armstrong: Hello, Dolly. When activated you will randomly see a lyric from Hello, Dolly in the upper right of your admin screen on every page. Version 1.7.2 | By Matt Mullenweg | View details. Enable auto-updates.
- My Database Admin**: Allows Read, Write, Update operations on database from Admin Dashboard. Version 1.1.23 | By wpshrike | View details. Enable auto-updates.
- Ninja Tables**: The Easiest & Fastest Responsive Table Plugin on WordPress. Multiple templates, drag-&-drop live table builder, multiple color scheme, and styles. Version 4.1.7 | By WPManageNinja LLC | View details. Enable auto-updates.
- TablePress**: Embed beautiful and feature-rich tables into your posts and pages, without having to write code. Version 1.14 | By Tobias Bäthge | View details | FAQ | Documentation | Support | Donate. Enable auto-updates.
- Ultimate Member**: The easiest way to create powerful online communities and beautiful user profiles with WordPress. Version 2.2.5 | By Ultimate Member | View details. Enable auto-updates.
- WP Htaccess Editor**: Safe and easy way to edit the .htaccess file directly from WP admin without using FTP. Version 1.70 | By WebFactory Ltd | View details | Plugin Homepage | Support | Rate the plugin ★★★★☆. Enable auto-updates.
- WP Popups Lite**: Beginner friendly WordPress popup builder plugin. Version 2.1.4.5 | By timersys | View details. Enable auto-updates.

At the bottom right, there is a link to 'Automatic Updates'.

[Figure 31-Route Plugins]

In the Trip-To-Hell CTF box, to implement the task number 09, created a S3 bucket with read only permissions. S3 bucket is not in a region it is available for globally.

The screenshot shows the AWS S3 console. The left sidebar includes 'Buckets', 'Access Points', 'Object Lambda Access Points', 'Multi-Region Access Points', 'Batch Operations', 'Access analyzer for S3', 'Block Public Access settings for this account', 'Storage Lens', 'Dashboards', 'AWS Organizations settings', 'Feature spotlight', and 'AWS Marketplace for S3'.

The main area displays an 'Account snapshot' with a 'View Storage Lens dashboard' button. Below is a table titled 'Buckets (2) Info' with a 'Create bucket' button. The table columns are Name, AWS Region, Access, and Creation date.

| Name | AWS Region | Access | Creation date |
|------------------|---|-------------------------------|--|
| myawsbucketth | Asia Pacific (Singapore) ap-southeast-1 | Objects can be public | September 20, 2021, 19:12:55 (UTC+05:30) |
| victorthevillain | US East (N. Virginia) us-east-1 | Bucket and objects not public | September 29, 2021, 23:48:25 (UTC+05:30) |

At the bottom, there is a search bar for 'Find buckets by name' and navigation controls. The footer includes links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'.

[Figure 32-S3 Bucket]

As the final level of the CTF project, created an account on tryhackme.com. After that created a room to host the CTF box and created 09 tasks related to the CTF box. In the first task players can join and visit the vulnerable website or they can easily google it.

The screenshot shows the TryHackMe web interface for managing a CTF room. On the left, a sidebar lists navigation options: General, Stats, Tasks (which is selected and highlighted in green), Design, Users, Clone, Access, and Delete. The main content area is titled "Description" and contains the following text:

```
B I U S X X, Ubuntu, Code 16, A, T!, 
```

You Need to Visit the www.travelmateth.com to complete this CTF Box.
 You are a FBI agent. You need to help the SWAT team to Catch a villain.
 Inspect the welcome page and click the link Then BooM.

Below this, under "Questions, Answers and Hints", there are three questions listed in a table:

| Question #1 | What is the Agents user name? | Answer | Hunter |
|-------------|--------------------------------------|--------|-----------------|
| Question #2 | What is Agents Password? | Answer | w!lh4Nt3r@fbi |
| Question #3 | What is the Given Key for the Agent? | Answer | 058726@RTR7!123 |

At the bottom of this section are two buttons: "+ Add more questions" and "Delete last question". Below the table are two more buttons: "Save" (green) and "Delete" (red).

[Figure 33-TryHackMe]

The screenshot shows a list of 9 tasks in a dark-themed interface. Each task is represented by a horizontal bar with a small icon and the task name. The tasks are:

- Task 1: The Inspector
- Task 2: Let the hunt begin
- Task 3: Pursuit of HappYness
- Task 4: Back To The Past 2
- Task 5: The Three Headed Serpent
- Task 6: Megalodon
- Task 7: Meet the Lucifer
- Task 8: Zap
- Task 9: MonaLisa

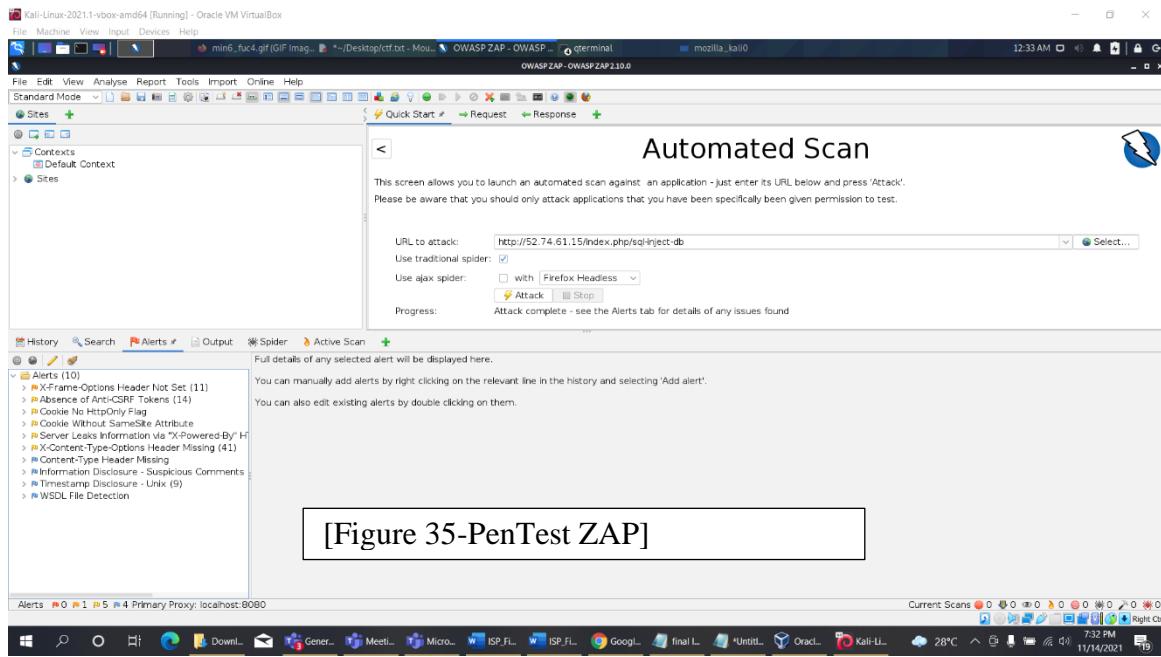
[Figure 34-TryHackMe]

2.4 Testing

As the first step after hosting the website, checked about its functions (ex: login function, register function, account maintaining). Those functions were successfully worked.

After the implementation of the cloud-based website, using security scanners the created web site was fully scanned. The main purpose of doing the security scan of the website was, needed to know is there any loopholes on the website than the created loopholes. By conducting the web scanning could analyze that there are no critical vulnerabilities except implemented ones. To penetrate the website used OWASP ZAP, Nmap, nikto, wapiti, skipfish, jSQL, scanQLi, SQLmap.

Scan results of OWASP ZAP



Nikto results:

```
[root@kali:~]# nikto -v2 -h http://44.199.17.110
[+] Target IP:   44.199.17.110
[+] Target Hostname: 44.199.17.110
[+] Target Port:  80
[+] Start Time: 2021-11-12 14:04:30 (GMT-5)

Server: Apache/2.4.48 (Ubuntu)
X-Frame-Options header: DENY
The X-XSS-Protection header is not present.
The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
Uncommon header 'link' found, with contents: <http://44.199.17.110/index.php/wp-json/>; rel='https://api.w.org/'.
Uncommon header 'rel=shortlink'.
The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
Uncommon header 'x-content-type-options' found, with contents: WordPress
Uncommon header 'x-xss-protection' found, with contents: 1
Uncommon header 'x-xss-protection' found, with contents: 1
OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
[...]
```

[Figure 36-PenTest Nikto]

Wapiti scan results:

Wapiti vulnerability report

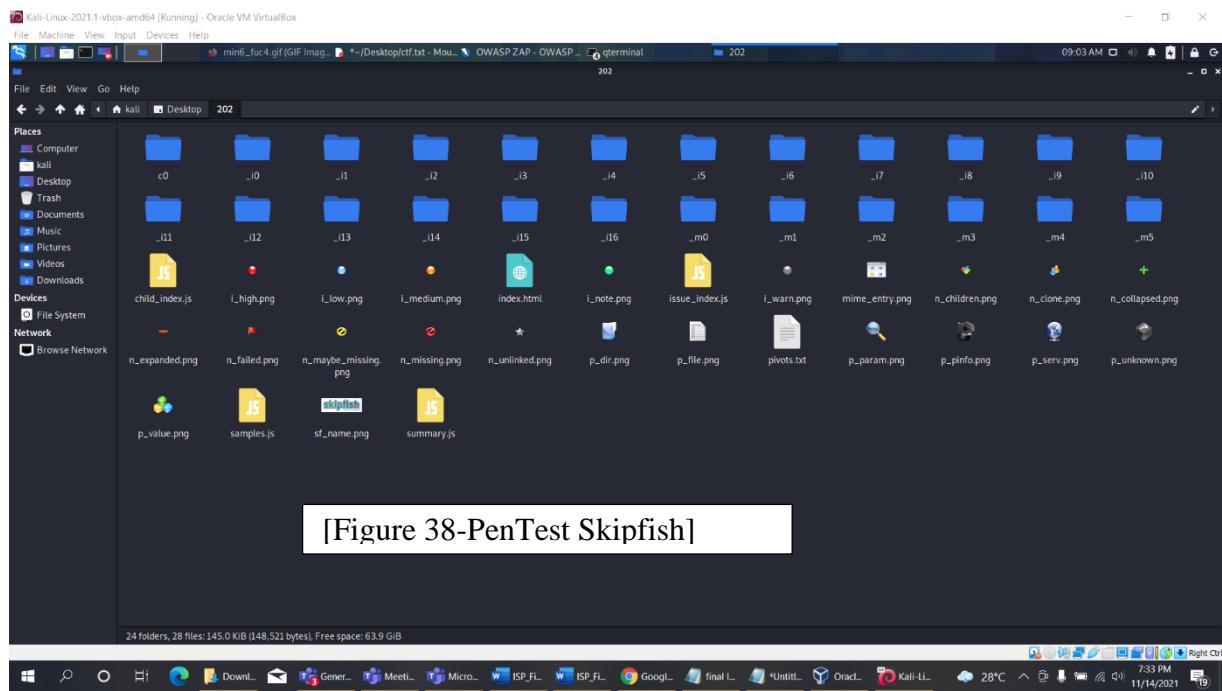
Target: <http://52.74.61.15/index.php/sql-inject-db/>

Date of the scan: Wed, 10 Nov 2021 15:48:40 +0000. Scope of the scan: folder

| Category | Number of vulnerabilities found |
|-----------------------------|---------------------------------|
| SQL Injection | 0 |
| Blind SQL Injection | 0 |
| File Handling | 0 |
| Cross Site Scripting | 0 |
| CRLF Injection | 0 |
| Commands execution | 0 |
| Httpaccess Bypass | 0 |
| Backup file | 0 |
| Potentially dangerous file | 0 |
| Server Side Request Forgery | 0 |

[Figure 37-PenTest Wapiti]

Skipfish scan result results:



[Figure 38-PenTest Skipfish]

3. Evaluation

3.1 Assessment of the Project results

When starting the CTF project the team had zero knowledge about cloud computing. To improve the knowledge, had to follow tutorials and practical. After gaining finite amount of knowledge about the AWS cloud platform and it uses the team decided to start the implementation of Trip-To-Hell CTF box.

As mentioned above in methodology sector the implementation of the instances, databases and other requirements was the main task of the CTF box. The backend of the CTF box was firstly developed due to the necessity of the security requirements and other requirements that need to implement the CTF levels.

When implementing the first task of the CTF box, tried to make that level easier because the players need to adapt to the vulnerable website. As the first step, implemented it with hiding a secret link address in the coding part of the website home page. The second task is dependent on the first task's findings. In second task, player must download audio file and analyze it with a tool called Deepsound. Inside the audio file there is an old link of a twitter post of the website. Player must capture that. That level is based on cryptography. Third level is different from above two levels. At that level player need to run a nikto scan on website URL to analyze it. Nikto will give some information about the website. By collecting that information player has to find a comment section which is hided by the administrator and capture the flag and other link. Inside that link there will be a password file. As the fourth level of the CTF player had to analyze the old twitter link that found on the second level with way-back machine. As the fifth level, implemented a brute force task. In that task player need to know how to use hydra and how to create input files for the hydra. In the sixth level of the Trip-To-Hell CTF the player needs to analyze Wireshark packets and collect the details. For that analysis, as a small help to the player, uploaded a txt file on the tryhackme.com to collect the data for the analysis. That level is based on network analysis and cryptography. In the seventh task using the knowledge about the basic cryptography player need to capture the level flag. In task number eight player will get information about a hidden web page from the previous task and using OWASP ZAP or any advance scanning tool the player needs to find the admin name and the web page created date (There will not be any critical vulnerability alerts. However, player has to analyze every information

specially the feed information that disclosed with low level vulnerability alerts.). The main intention to implement this level is to mention that even with the low-level information disclosures that can be critical information leakages happened.

Task number nine is implemented with cloud computing basics. To capture the flag of this MonaLisa level player need to have basic knowledge about cloud computing. In this level player has to connect to the S3 via connecting EC2 instance with its DNS address and download an image file from the read only S3 bucket and analyze it with STool (steganography tool) and capture the final flag. In this level player will be needed to use Nmap tool too. Because he or she will need to find the open port details of the EC2 instance to connect to it.

With these backend, frontend web and CTF level implementations the team could successfully complete the CTF project. However, there were some failures happened when coming through this CTF building project. When trying to set the route 53 domain on the vulnerable website the web site was completely crashed due to misconfigurations. To overcome this issue, had to re-create the entire vulnerable website again. When creating the RDS database inside the Singapore region the MySQL database was unavailable due to technical failures of the Amazon server. After few research about the AWS server, found out that the Singapore region server face to technical failures commonly. Therefore, the RDS database region was changed to the North Virginia region and created it again to overcome that issue. In the testing phase, while doing the penetration testing on the created Travel Mate website, a system crash happened. jSQL tool was used at that time to pentest the web page to analyze for sql injection vulnerabilities. The penetration went wrong, and the web system completely crashed. However, after restarting the servers and databases the error was fixed by the system automatic repairing method. While configuring the permalinks on the website using WordPress that leaded to a system failure with website functioning errors. To resolve that the configuration details of the permalinks had to reset by manually.

3.2 Lessons Learned

By doing this CTF building project could gain knowledge about cloud computing basics and could learn about web site hosting on AWS platform. Also learned about the uses of WordPress and its security features. By hosting a web site on cloud, could learn number of security vulnerabilities and misconfigurations that can happen due to the lack of knowledge about security and configurations.

When configuring the latest version of WordPress with database found out that to install plugins the admin needs to give read, write, execute permission for wp-content directory and plugins folder. After doing research about it, could figure out that it is a kind of vulnerability that is only in the WordPress newest (10/11/2021) version 5.8.1.

3.3 Future Work

Expecting to fine tune the present version of the Trip-To-Hell CTF box and build an image file that players can host the website locally. However, the local hosting part might come out as an extended version of current CTF box. As a team, that plan to continue this CTF box with more features in future.

4. Conclusion

The CTF project created with web-based system that based on cloud computing. To implement the CTF box used AWS platform as cloud service provider and used WordPress as the website development tool. RDS Database and the EC2 instance of the website was connected with WordPress and with all these mechanisms developed a fully function vulnerable website. CTF box levels are implemented and based on the vulnerable Travel Mate website. There is main nine levels in the created CTF box. Try Hack Me website was selected to host this Trip-To-Hell CTF box.

5. References

[01] YouTube: <https://www.youtube.com/watch?v=ulprqHHWlNg&t=7963s>

[02] WordPrss: <https://wordpress.org/>

[03] Try Hack Me: <https://tryhackme.com/dashboard>

[04] Contrast: <https://www.contrastsecurity.com/security-influencers/tips-tactics-ctf-event>

Appendix A: Test Results

Link to the walkthrough video: https://drive.google.com/drive/folders/1GR-c9JeRazk2N1wVwWXu2U_ORVsSZvcp?usp=sharing