Round 2 - Project Building(Cloud-Honeypot-Project in cloud computing)

- **Detailed Project Plan: Detailed Project Plan for Cloud-Honeypot-Project (Vultr)**
- **Project Objectives**

The Cloud Honeypot Lab aims to build a safe and isolated environment in the cloud that mimics a real enterprise network, designed to attract and study cyberattacks. By deploying this honeypot, we intend to observe, analyze, and understand the behavior of attackers, enabling proactive defense strategies. This project will help in gaining hands-on experience with setting up cloud-based security solutions, enhancing skills in network security, and log analysis.

Key Objectives:

- Create a cloud-based honeypot environment that attracts and logs cyberattacks.
- Capture real-time data on attacks for analysis and pattern recognition.
- Build dashboards for visualizing attack patterns using tools like Kibana.
- Ensure security of the host machine by isolating the honeypot in a cloud environment (using Vultr).
- Develop skills in cloud computing, firewall configuration, and security monitoring.

2. Detailed Timeline with Milestones

Phase	Milestone	Tasks	Timeline
Phase 1: Planning	Project Kickoff	- Define project scope and objectives - Set up Vultr account	Week 1
Phase 2: Environment Setup	Cloud VM Setup and Configuration	- Create a new instance with Debian 12 - Configure firewall settings	Week 2
Phase 3: Honeypot Deployment	Tpot Installation	- Install Tpot on Debian - Setup user authentication for Tpot interface	Week 3
Phase 4: Security Configuration	Firewall Reconfiguration	- Allow inbound connections to Tpot web interface	Week 4

Phase	Milestone	Tasks	Timeline
Phase 5: Data Collection	Honeypot Observation & Logging	- Run honeypot to collect data - Monitor attack map and Kibana dashboards	Weeks 5-6
Phase 6: Analysis & Reporting	Data Analysis & Final Report	- Analyze collected logs - Generate visual reports using Kibana	Week 7
Phase 7: Project Closure	Project Review & Documentation	- Review project objectives - Complete documentation and presentation	Week 8

3. Description of Deliverables

A. Project Documentation

- **A comprehensive document that covers the project overview, objectives, tools used, setup processes, and lessons learned.**
- **Includes step-by-step instructions on setting up the honeypot environment in the cloud.**
- **Contains screenshots and references (as seen in images [1], [2], [3], and [4]) to demonstrate successful implementation.**

B. Cloud Honeypot Environment

- **A fully functional honeypot deployed on the Vultr cloud platform using Debian 12 and Tpot.**
- **The environment will be configured to capture and log incoming cyberattacks while ensuring the host machine remains safe.**

C. Attack Map Dashboard (Tpot)

- **A real-time attack visualization dashboard showing the geographic locations of incoming attacks.**
- **The attack map will provide insights into the source, frequency, and type of attacks targeting the honeypot.**

D. Kibana Dashboard & Analysis

- **A Kibana dashboard set up to visualize and analyze the data collected by the honeypot.**
- **The dashboard will include:**

  - **Honeytrap Overview: Summary of attacks, including source IPs, ports targeted, and attack types.**
  - **Event Timeline: Timeline of events showing when attacks occurred.**
  - **Patterns and Trends: Visualization of attack patterns and potential indicators of compromise (IoCs).**

E. Final Report

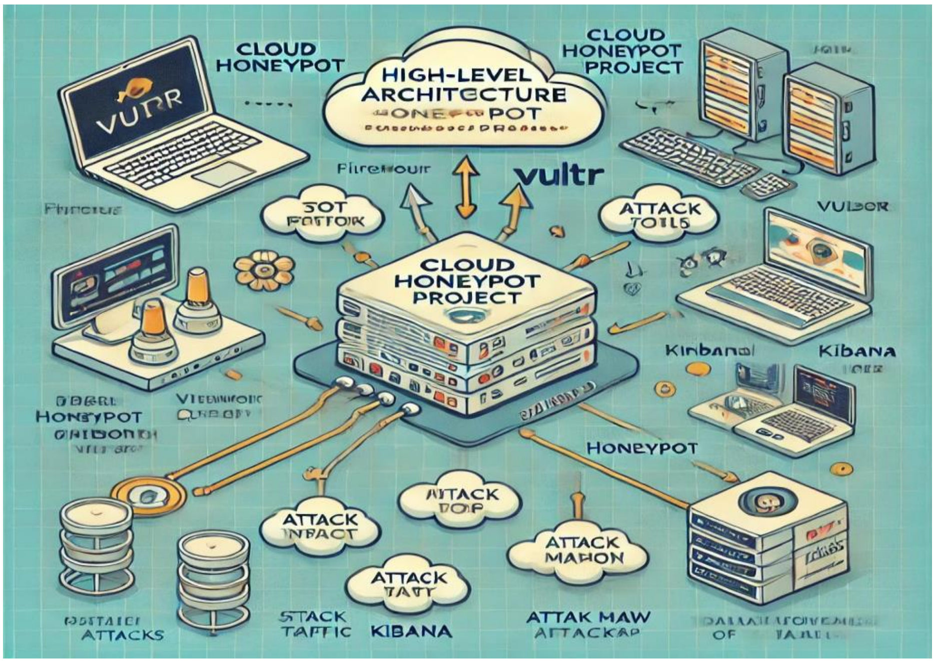
- **A detailed report summarizing the findings from the honeypot data analysis.**
- **Includes:**
  - **Key Observations: Highlights of significant attacks or patterns.**
  - **Recommendations: Suggestions for improving network security based on observed attack behavior.**
  - **Future Work: Potential enhancements for the honeypot setup and areas for further research.**

➢ **Architecture Diagrams:**

High-Level Architecture

The high-level architecture diagram should depict the overall structure of your honeypot setup in the cloud environment. This will include the following components:

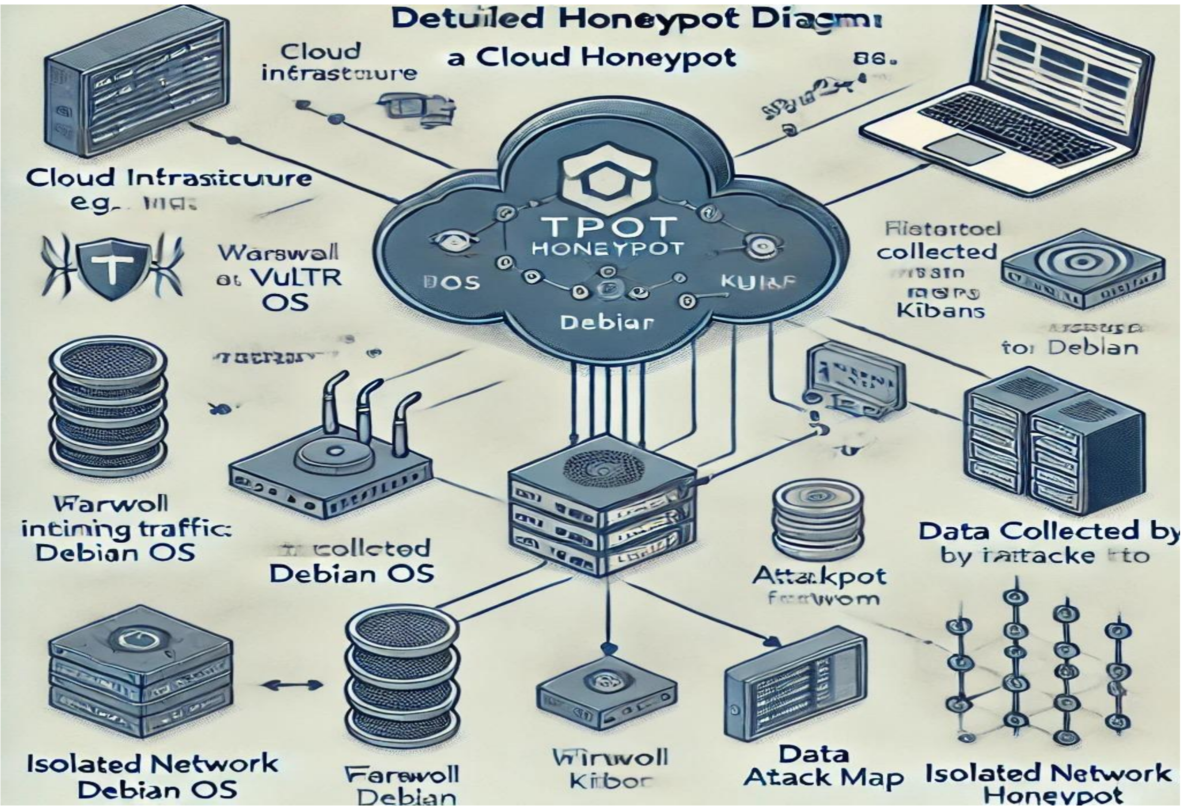
➢ **High-level architecture diagram**



- **Vultr Cloud Platform**: The main hosting environment for your honeypot.
- **Virtual Machine (VM)**: Deployed with a Debian 12 OS.
- **Firewall Configuration**: Controls incoming and outgoing traffic to secure the environment.

- **Tpot Honeypot**: The primary software used to mimic an enterprise network for attackers. It includes features like Attackmap and Kibana Dashboard.
- **Web Interface**: Used for monitoring and managing the honeypot, accessed via a specified port.

➢ **Detailed component diagrams :**

- 

  The diagram would show how all these elements interact, including how the firewall controls access to the honeypot and how data flows from the Tpot honeypot to monitoring tools like Kibana.

  Detailed Component Diagrams

  The detailed component diagram should break down the specific configurations of the setup:

1. **Cloud VM Configuration**:
   - **Debian OS**: Installed on the Vultr cloud instance.
   - **User Management**: Root user and additional user with sudo privileges.
   - **Network Security**: Firewall settings configured to block or allow specific ports.
2. **Tpot Honeypot Components**:
   - **Attackmap**: Visual representation of incoming attacks.
   - **Kibana Dashboard**: Interface for log data analysis.
   - **Honeytrap**: Modules capturing different types of attacks.

   This diagram can show the internal structure of the Tpot setup, including the interfaces and ports used.

   Network Topology (if applicable)

The network topology diagram will illustrate the network setup for your honeypot environment:

- **Cloud Network**: Shows the Vultr cloud environment where the honeypot is hosted.
- **Firewall**: Positioned between the public internet and your cloud VM to filter traffic.
- **External Attackers**: Represented as potential threats from the internet targeting your honeypot.
- **Data Flow**: Displays how attack data is collected by Tpot and visualized through Kibana.

These diagrams collectively provide a clear view of how your honeypot system is architected, secured, and monitored.

➢ **Technical Documentation:**

High-Level Architecture Diagram

The high-level architecture diagram should depict the overall structure of your honeypot setup in the cloud environment. This will include the following components:

- **Vultr Cloud Platform**: The main hosting environment for your honeypot.
- **Virtual Machine (VM)**: Deployed with a Debian 12 OS.
- **Firewall Configuration**: Controls incoming and outgoing traffic to secure the environment.
- **Tpot Honeypot**: The primary software used to mimic an enterprise network for attackers. It includes features like Attackmap and Kibana Dashboard.
- **Web Interface**: Used for monitoring and managing the honeypot, accessed via a specified port.

The diagram would show how all these elements interact, including how the firewall controls access to the honeypot and how data flows from the Tpot honeypot to monitoring tools like Kibana.

Detailed Component Diagrams

The detailed component diagram should break down the specific configurations of the setup:

1. **Cloud VM Configuration**:
    - **Debian OS**: Installed on the Vultr cloud instance.
    - **User Management**: Root user and additional user with sudo privileges.
    - **Network Security**: Firewall settings configured to block or allow specific ports.
2. **Tpot Honeypot Components**:
    - **Attackmap**: Visual representation of incoming attacks.
    - **Kibana Dashboard**: Interface for log data analysis.
    - **Honeytrap**: Modules capturing different types of attacks.

This diagram can show the internal structure of the Tpot setup, including the interfaces and ports used.

Network Topology (if applicable)

The network topology diagram will illustrate the network setup for your honeypot environment:

- **Cloud Network**: Shows the Vultr cloud environment where the honeypot is hosted.
- **Firewall**: Positioned between the public internet and your cloud VM to filter traffic.
- **External Attackers**: Represented as potential threats from the internet targeting your honeypot.
- **Data Flow**: Displays how attack data is collected by Tpot and visualized through Kibana. These diagrams collectively provide a clear view of how your honeypot system is architected, secured, and monitored.

GitHub Repository:

- **Source Code**: The project's source code should contain the configuration files, scripts for setting up the Tpot honeypot, firewall settings, and any custom scripts used in the setup.
- **Build and Deployment Instructions**: Include detailed steps on how to deploy the environment. This could involve:
  - **Vultr Setup**: How to create a VM instance using Vultr and configure it.
  - **Tpot Installation**: Instructions for setting up Tpot, including the steps to run the installation script and configure the web interface.
  - **Firewall Configuration**: How to secure the environment by adjusting firewall rules to block unwanted traffic and allow specific connections.
  - **Kibana & Attackmap**: Instructions for accessing the Kibana dashboard and Attackmap to analyze the incoming attack data.

You can create a GitHub repository for your project and host the full codebase and documentation there. Here's a structure outline for your repository:

bash

Copy code

```
Cloud-Honeypot-Project/
│
├── README.md               # Project overview and instructions
├── deployment/
│   ├── setup_vultr.sh        # Script to set up the Vultr instance
│   └── tpot_installation.sh    # Script to install Tpot honeypot
├── firewall_config/
│   └── firewall_setup.sh       # Script to configure the firewall
├── config/
│   ├── tpot_config.yaml        # Tpot honeypot configuration
│   └── kibana_config.json      # Kibana dashboard configuration
├── logs/
│   ├── attackmap_logs.json     # Sample attack map data
│   └── kibana_dashboard_data.json # Sample Kibana logs
└── images/
```

```
 ├── tpot_installed.png      # Screenshot of Tpot installed
└── kibana_dashboard.png    # Screenshot of Kibana dashboard
```

This repository will provide a working prototype for others to replicate the honeypot setup and utilize your work.

Feel free to upload your project to GitHub and share the repository link for further assistance with build and deployment.

GitHub Repository:

- **Source Code**: The project's source code should contain the configuration files, scripts for setting up the Tpot honeypot, firewall settings, and any custom scripts used in the setup.
- **Build and Deployment Instructions**: Include detailed steps on how to deploy the environment. This could involve:
  - **Vultr Setup**: How to create a VM instance using Vultr and configure it.
  - **Tpot Installation**: Instructions for setting up Tpot, including the steps to run the installation script and configure the web interface.
  - **Firewall Configuration**: How to secure the environment by adjusting firewall rules to block unwanted traffic and allow specific connections.
  - **Kibana & Attackmap**: Instructions for accessing the Kibana dashboard and └── kibana_dashboard.png # Screenshot of Kibana dashboard

This repository will provide a working prototype for others to replicate the honeypot setup and utilize your work.

Feel free to upload your project to GitHub and share the repository link for further assistance with build and deployment.

Link : https://github.com/rajat738/Cloud-Honeypot-Project.git