Sir Syed University of Engineering & Technology (SSUET)

Department of Software Engineering

# Cyber Security Program

***Course Name: Digital Forensics (CY-301L)***

# PROJECT REPORT

## PROJECT TITLE

## Web HoneyPot Forensics With Reverse Shell

****

**Muhammad Hasnain**

**OBJECTIVE:**

The primary objective of this project is to design and implement a web-based honeypot capable of capturing and analyzing reverse shell attacks in a controlled environment. This helps understand how attackers exploit vulnerable web applications to gain unauthorized access to systems and how forensic techniques can be used to detect and analyze such attacks.

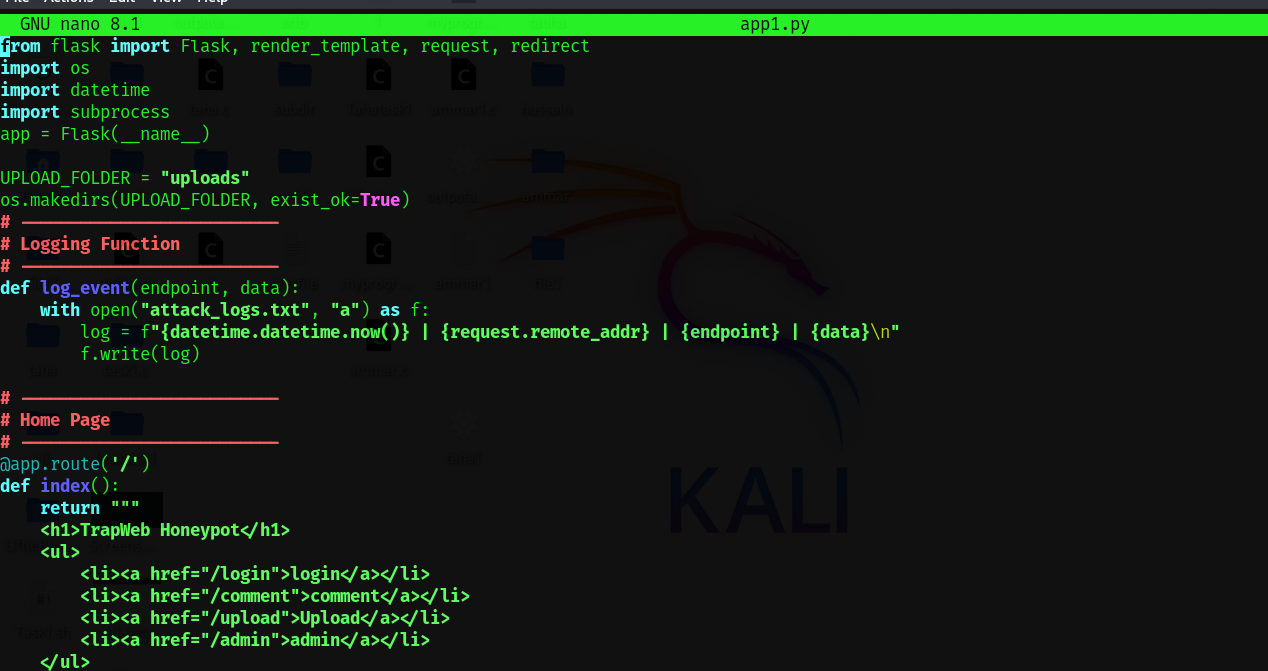
Specifically, this project aims to:

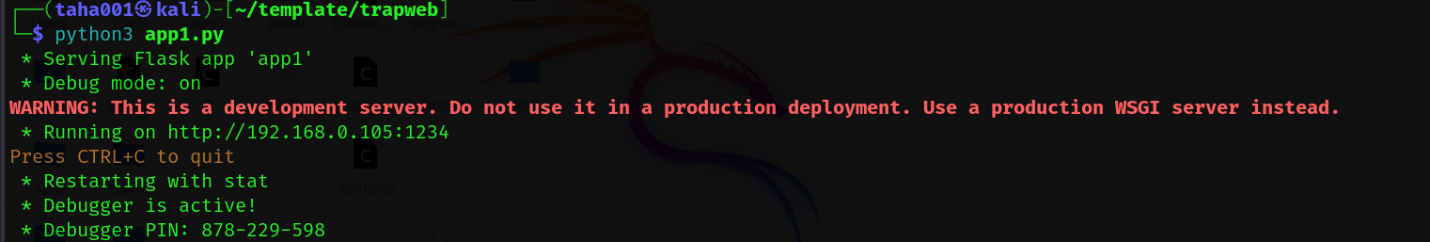
* Deploy a web honeypot server to attract attackers.
* Simulate a reverse shell attack from a second machine to the honeypot.
* Analyze and document the attack steps and evidence.

**WALKTHROUGH**

**Step 1: Created the Honeypot Server**

We first created a basic honeypot server using **Python Flask** framework. The server acted as a vulnerable web application that attackers could interact with.

**

**

**Step 2: Developed a Web Page with Log Feature**

We then developed a simple web page hosted on the honeypot server. This page included a log feature designed to record activities such as file uploads and suspicious requests.

The purpose of this log was to track:

* IP addresses of visitors.
* Uploaded files and associated metadata.
* Access attempts to restricted resources.

A white and blue screen

AI-generated content may be incorrect.

**Step 3: Uploaded Reverse Shell from Second Machine**

From a second machine on the same local network (both machines running Kali Linux), we created and uploaded a **reverse shell script** to the honeypot web page through its upload feature.

The reverse shell file was crafted to connect back to the second machine, providing remote command-line access to the honeypot server.A screenshot of a computer

AI-generated content may be incorrect.

**Step 4: Gained Access to Honeypot Terminal**

Once the reverse shell was successfully uploaded and executed, we received a reverse connection from the honeypot server back to the attacker's machine.

This allowed us to:

* Gain shell access to the honeypot (first machine) terminal from the second machine.
* Execute arbitrary commands remotely on the honeypot.
* Simulate an actual compromise scenario.

A screenshot of a computer

AI-generated content may be incorrect.

Both machines were connected to the same network, which is critical for establishing the reverse shell connection.

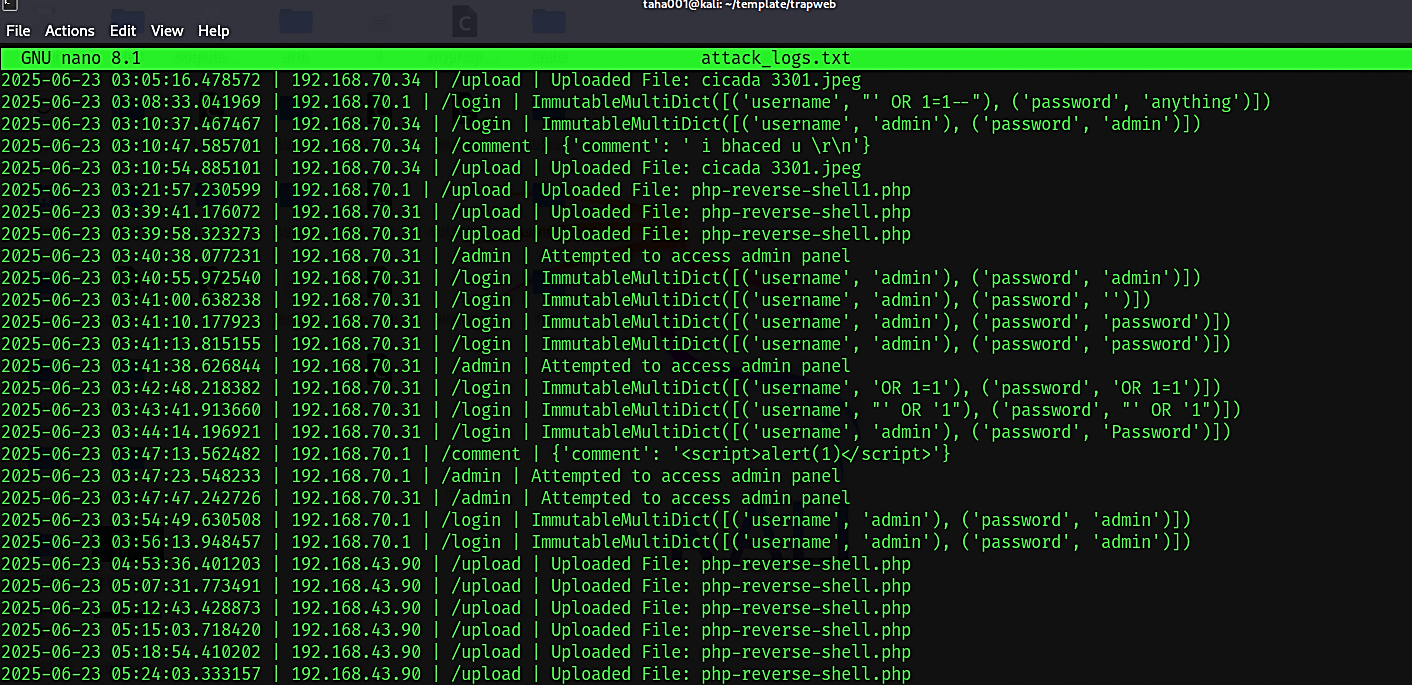
**Step 5: Captured Attacker's Information on Honeypot (First Machine)**

After gaining shell access from the second machine, we verified that the honeypot server (first machine) successfully logged the attacker's information.

Specifically, the honeypot collected and recorded:

* Attacker’s IP address: The originating IP of the second machine was captured in the server logs.
* Uploaded file details: The exact filename, upload time, and file path were logged for forensic analysis.
* Access logs: All HTTP requests, including suspicious activity and command execution attempts, were stored.

This step demonstrates the effectiveness of a properly configured honeypot in tracing attacker behavior and gathering digital evidence. The captured data can be used for further investigation, correlation with other incidents, and legal or academic reporting.



**CONCLUSION**

Through this project, we successfully demonstrated how a web honeypot can be exploited using a reverse shell attack and how such attacks can be monitored and analyzed for forensic purposes. The logs and activity trails recorded during the process provide valuable insight for incident response and further security improvements.