🛡️ Real-Time Honeypot System – Project Report

# 📌 Project Title

Fake FTP Honeypot with Intrusion Detection using Python

# 🎯 Objective

To create a deceptive FTP server that attracts potential attackers, logs their commands, and captures their webcam snapshot on suspicious activities — for cybersecurity awareness and analysis.

# 🧰 Tech Stack

• Python  
• Socket Programming  
• SQLite (for command logging)  
• OpenCV (for webcam snapshots)  
• Windows 11 (Local Setup)

# ⚙️ System Components

1. Honeypot Server (server.py): Simulates a fake FTP login, logs commands, and captures webcam snapshot on suspicious keywords.  
2. Client Simulator (client.py): Simulates an attacker interacting with the honeypot.  
3. Logger (logger.py): Logs all commands to a SQLite database.  
4. Webcam Capture (webcam.py): Captures webcam snapshots using OpenCV.

# 🧪 Testing Procedure

• Started the honeypot using server.py  
• Connected using client.py  
• Typed suspicious commands like 'whoami', 'cat /etc/passwd', 'wget'  
• Verified that commands were logged and webcam snapshots were taken

# 📁 Database Schema

Table: logs  
Fields: id, ip, command, timestamp

# 🔒 Security Awareness Goal

This project educates users on how real systems can detect and respond to intrusion attempts. Helps simulate how cyber attacks can be monitored and recorded silently.

# 🌐 Future Improvements

• Deploy on a LAN or public IP  
• Add email alerts for specific command triggers  
• Use machine learning for command pattern analysis  
• Integrate with dashboards like ELK or Grafana for log visualization

# ✅ Conclusion

This honeypot system is a lightweight yet powerful tool for understanding how attackers interact with fake systems. It helps raise awareness, test response strategies, and collect valuable threat intelligence in a controlled environment.