Project Report: SSH Botnet Simulation

1. Objective

The objective of this project is to simulate a lightweight SSH-based botnet framework in a controlled lab environment for educational and red-team training purposes. The simulation demonstrates how a central Command-and-Control (C2) server can remotely manage multiple SSH-compromised hosts ("bots"), execute arbitrary commands, and launch coordinated simulated attacks such as SYN flood.

2. Project Overview

This project mimics real-world adversarial behavior, showcasing how botnets communicate with infected hosts via the SSH protocol. Unlike malware-based botnets that use HTTP, IRC, or custom binary protocols, this simulation leverages secure shell (SSH) sessions to achieve persistent, encrypted command execution.

The framework consists of:

- A **C2 controller** that connects to multiple bots via SSH.
- A **persistent JSON-based storage** system to maintain bot credentials and host information.
- An **interactive shell interface** for issuing live commands to selected bots.
- A **SYN flood attack module** that simulates distributed denial-of-service (DDoS) behavior using crafted TCP packets.

3. Key Features

✅ Botnet Controller

- Connects to multiple SSH-enabled systems.
- Authenticates using credentials from a JSON file.
- Tracks connection status for each bot.

✅ Remote Command Execution

- Supports single-command dispatch to all or selected bots.
- Returns real-time output from each bot terminal.

✅ Interactive Shell

- User can drop into an interactive shell session with any connected bot.
- Commands are executed as if typed in the bot's terminal.

✅ SYN Flood Attack Simulation

- Uses `Scapy` to simulate SYN flood attacks from any bot to a target IP and port.
- Demonstrates packet crafting, spoofing, and multi-threaded flood techniques.

4. Technology Stack

Technology	Purpose
Python 3	Core programming language
`paramiko`	SSH protocol handling
`Scapy`	Packet crafting and SYN flood simulation
`socket`, `json`, `threading`	Core Python modules for I/O and concurrency

5. Ethical Consideration

This project was conducted in a **controlled and isolated lab environment**, with all bots hosted on virtual machines with preconfigured SSH services. No external systems were targeted. The project adheres to **ethical hacking guidelines** and is intended purely for educational and research use.

6. Learning Outcomes

- Deep understanding of **SSH authentication, command dispatching, and session management**.
- Hands-on practice in **multithreading, JSON data handling, and socket programming**.
- Familiarity with **DDoS attack mechanics**, particularly SYN flood attacks and how they can be simulated safely.
- Implementation of **C2 architecture**, which mirrors real-world offensive frameworks used by red teamers and adversaries.

7. Conclusion

The SSH Botnet Simulation project was a successful demonstration of a core offensive security concept: **remote host control and orchestration via SSH**. It showcased the complete lifecycle of botnet operation — including setup, persistence, command execution, and attack simulation — while remaining ethical and confined to a virtual testbed.

This project enriched my understanding of offensive tools, command-and-control strategies, and the importance of securing SSH services against brute-force and credential-stuffing attacks.