SSH Botnet Using Python

By Inlighn Tech

Objective:

This project demonstrates how an **SSH-based botnet** can be built and controlled using Python. The script allows users to connect to multiple SSH-enabled devices (bots), execute remote commands, and even launch a basic **DDoS attack**. The primary goal of this project is to **educate students about cybersecurity threats, botnet structures, and SSH exploitation techniques.**

Project Overview:

This Python script provides the following functionalities:

1. **List Bots:** View all connected SSH bots.
2. **Execute Commands:** Send commands to all bots for remote execution.
3. **Interactive Bash Shell:** Run commands interactively on bots.
4. **Add Bots:** Add new SSH clients (bots) to the botnet network.
5. **Perform a DDoS Attack:** Simulate a SYN flood attack on a target.
6. **Save & Load Botnet:** Store bot information in a JSON file to reconnect later.

How the Project Works:

1. **Connecting to SSH Bots:**
    a. **Uses `pxssh` (from `pexpect`) to establish SSH sessions with remote systems.**
    b. **Stores the SSH credentials and sessions in a list (`botnet`).**
2. **Executing Remote Commands:**
    a. **Sends commands via SSH and collects the output from each bot.**
    b. **Displays results for each bot separately.**
3. **Interactive Bash Shell:**
    a. **Opens a bash shell that allows real-time execution of commands on bots.**
    b. **Useful for testing commands across multiple machines.**
4. **Botnet Management:**
    a. **Bots are stored in a `botnet.json` file, allowing persistent connections.**
    b. **Automatically reconnects to previously added bots.**

5. **DDoS Attack (SYN Flood):**
    a. **Uses `Scapy` to generate SYN packets to flood a target.**
    b. **Demonstrates the impact of a SYN flood attack on network resources.**

Key Concepts Covered:

- **Botnet Structure:** How botnets operate using SSH connections.
- **Remote Command Execution:** Running commands on multiple remote machines.
- **Persistence in Botnets:** Storing and reconnecting to bots automatically.
- **Networking and Security:** Understanding how SYN floods affect a target.
- **Ethical Hacking:** Recognizing vulnerabilities in exposed SSH servers.

Step-by-Step Implementation:

1. **Connecting to SSH Bots**
    a. Uses SSH authentication to log in to remote machines.
    b. Maintains a session object for command execution.
2. **Sending Commands to Bots**
    a. Iterates through all bots and sends commands via SSH.
    b. Displays the results in a structured manner.
3. **Interactive Bash Shell**
    a. Allows users to execute commands interactively on all bots.
    b. Each command is sent and executed remotely.
4. **DDoS Simulation**
    a. Generates large volumes of **SYN packets** using `Scapy`.
    b. Floods a target server to demonstrate network resource exhaustion.
5. **Persistent Botnet Storage**
    a. Saves bot credentials in a JSON file for later use.
    b. Loads saved bots at startup and reconnects automatically.

Expected Outcomes:

By completing this project, students will:

- Understand **how SSH can be exploited** for botnet operations.
- Learn **how botnets execute remote commands** efficiently.
- Gain knowledge about **network attacks like SYN flooding.**
- Develop hands-on experience in **ethical hacking and penetration testing.**

- Learn how **attackers use SSH vulnerabilities** and how to **defend against them.**

Next Steps:

Students should implement their own version of the SSH Botnet using the outlined concepts. A video tutorial will be provided later to demonstrate the correct implementation and security countermeasures.

For further enhancements, students can:

- **Implement Encryption: Secure SSH communication using asymmetric encryption.**
- **Add Persistence: Ensure bots automatically reconnect after a system reboot.**
- **Expand DDoS Capabilities: Integrate UDP and ICMP flooding techniques.**
- **Botnet Evasion Techniques: Implement stealth mechanisms to avoid detection by security tools.**
- **Anti-Botnet Defense: Develop a detection tool to identify SSH botnet activity in a network.**

Disclaimer: This project is for educational and ethical hacking purposes only.

Unauthorized access to systems without consent is illegal and violates cybersecurity laws. Always obtain permission before testing these techniques on any system.