### DOCUMENTATION ON BASIC SCRIPTING TASK 2. Create a Python script to:Parse logs from a web server.Identify IPs causing the most failed login attempts.Block those IPs by dynamically updating firewall rules.

**Objective:**  
Develop a Python script to monitor web server logs, identify failed login attempts, and dynamically block offending IP addresses using firewall rules.

### Step 1: Set Up the Environment

1. **Ensure Prerequisites**:
   * Python 3 is installed.
   * Install wordpress.
   * Access to server logs (e.g., Apache logs in /var/log/apache2/access.log).
   * Install wordpress plugin : Fail2ban (to monitor unknown user in auth.log)
   * Appropriate permissions to modify firewall rules (requires sudo).
2. **Install Required Python Libraries**: If additional libraries like re or subprocess are needed:

pip install re

pip install subprocess

### **Step 2: Create the Python Script**

import re  
import subprocess  
LOG\_FILE = "/var/log/auth.log"  # Path to the log file  
THRESHOLD = 5  # Number of failed attempts before blocking  
BLOCKED\_IPS\_FILE = "/var/log/blocked\_ips.txt"  # File to keep track of blocked Ips  
def get\_failed\_login\_ips():  
    """Parse the log file and return a dictionary of IPs with failed login attempts."""  
    failed\_ips = {}  
    with open(LOG\_FILE, "r") as file:  
        for line in file:  
            ip = line.split()[0]  # Extract IP address            # Detect failed login attempts due to "unknown user"  
            if "wp-login.php" in line and "unknown user" in line:  
                failed\_ips[ip] = failed\_ips.get(ip, 0) + 1            # Additional checks for "invalid username"  
            elif "wp-login.php" in line and "invalid username" in line:  
                failed\_ips[ip] = failed\_ips.get(ip, 0) + 1    return failed\_ipsdef block\_ip(ip):  
    """Block an IP using iptables."""  
    try:  
        subprocess.run(["sudo", "iptables", "-A", "INPUT", "-s", ip, "-j", "DROP"], check=True)  
        print(f"Blocked IP: {ip}")  
        with open(BLOCKED\_IPS\_FILE, "a") as file:  
            file.write(ip + "\n")  
    except subprocess.CalledProcessError as e:  
        print(f"Error blocking IP {ip}: {e}")def main():  
    """Main function to monitor logs and block offending IPs."""  
 failed\_ips = get\_failed\_login\_ips()    # Read already blocked IPs to avoid reblocking  
    try:  
        with open(BLOCKED\_IPS\_FILE, "r") as file:  
            blocked\_ips = set(file.read().splitlines())  
    except FileNotFoundError:  
        blocked\_ips = set()    # Check and block IPs exceeding the threshold  
    for ip, count in failed\_ips.items():  
        if count >= THRESHOLD and ip not in blocked\_ips:  
            block\_ip(ip)if \_\_name\_\_ == "\_\_main\_\_":  
    main()  
  
\*\* Change Permissions   
chmod +x block\_failed\_logins.py

### **Step 3: Perform failed login attempts to wordpress website by giving wrong username and password**

1. Run the script manually to check its functionality:

sudo python3 block\_failed\_logins.py

1. Verify that offending IPs are blocked using:

sudo iptables -L

### **Step 4: Automate the Task with a Cron Job**

1. Open the crontab editor:

crontab -e

1. Add the following line to run the script every 5 minutes:  
     
   \*/5 \* \* \* \* /usr/bin/python3 /home/ubuntu/block\_failed\_logins.py  
     
   Save and exit.

