This guide will walk you through integrating your Cowrie SSH honeypot with Slack using a Python script that sends alerts in real-time when specific honeypot activity occurs.

✅ Prerequisites

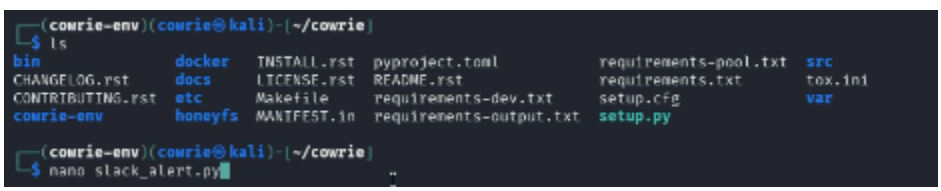
- Cowrie is already installed and running (see the Cowrie Install Guide)
- A Slack workspace with permissions to create an app

---

1. Create a Python Script

Make sure you are inside your `cowrie` directory:

cd ~/cowrie
nano slack_alerts.py



Paste the contents of the provided slack_alerts.py script. You can find this file in the GitHub repo or use the template in this project.

Before saving, update the following variables:

- Replace `WEBHOOK_URL` with your Slack Incoming Webhook URL
- Make sure `LOG_FILE` points to your Cowrie log file (e.g., `/home/cowrie/cowrie/var/log/cowrie/cowrie.log`)

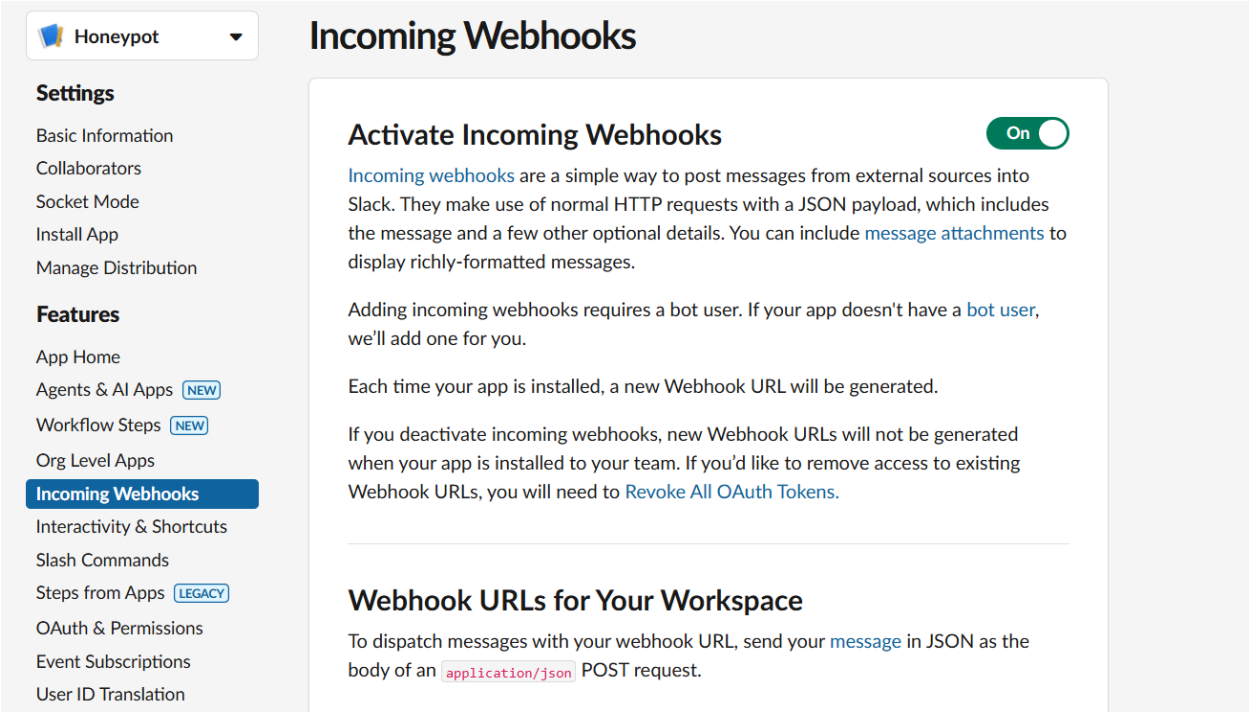
---

2. Set Up Slack Webhook

1. Go to https://api.slack.com/apps

2. Click **Create New App** > **From Scratch**
3. Name it and select your workspace
4. Select **Incoming Webhooks** from the features list
5. Toggle to enable Incoming Webhooks
6. Click **Add New Webhook to Workspace**
7. Choose a Slack channel to receive notifications
8. Copy the **Webhook URL**

Paste this URL into the `WEBHOOK_URL` field in your script.



```
cowrie@kali: ~/cowrie
File  Actions  Edit  View  Help
  GNU nano 8.3                    slack_alert.py *
import json
import requests
import time
import os

# ✅ Define your Slack webhook URL here
WEBHOOK_URL = ''

# ✅ Define the correct path to Cowrie's log file
LOG_FILE = '/home/cowrie/cowrie/var/log/cowrie/cowrie.log'

def send_to_slack(message):
    payload = {'text': message}
    headers = {'Content-Type': 'application/json'}
    try:
        requests.post(WEBHOOK_URL, data=json.dumps(payload), headers=headers)
    except Exception as e:
        print(f"Slack send failed: {e}")

def follow(logfile):
    logfile.seek(0, os.SEEK_END)
    while True:
        line = logfile.readline()
        if not line:
            time.sleep(0.2)
            continue

^G Help      ^O Write Out  ^F Where Is   ^K Cut      ^T Execute   ^C Lo
^X Exit      ^R Read File  ^\ Replace    ^U Paste    ^J Justify   ^/ Go
```

Once you have the script and make sure that your webhook url and log file are correct we will save the file

---

3. Save and Run the Script

Save the script and make it executable (if necessary):

chmod +x slack_alerts.py
python3 slack_alerts.py

```
┌──(cowrie-env)(cowrie㉿kali)-[~/cowrie]
└─$ cd /home/cowrie/cowrie
python3 slack_alert.py
```

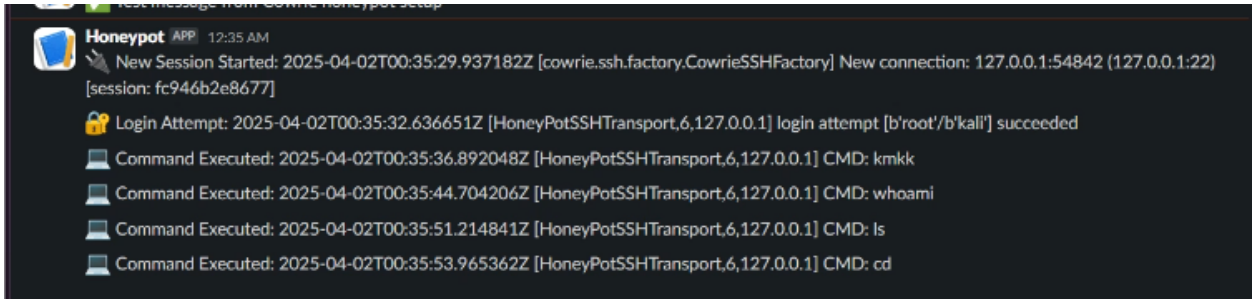
Leave this script running in a terminal or consider running it as a background process using `tmux` or `screen`.

---

4. Test the Integration

Trigger some activity in your honeypot:

ssh cowrie@localhost



Try running some commands or just logging in with random passwords. You should see messages in your Slack channel like:

- 🔐 Login Attempt
- 💻 Command Executed
- 📥 File Download Attempt
- 🔌 New Session Started
- 🔚 Session Ended

🎉 Success!

You've successfully integrated Slack with Cowrie, yayyy!! Every time an attacker interacts with your honeypot, Slack will notify you in real-time, allowing you to monitor attacks as they happen.

> Tip: You can extend this integration to send alerts to other platforms like Discord, email, or log to a database.

📌 Next Steps

- Add support for log rotation
- Log alerts to a file for offline analysis
- Build a dashboard with ELK, Wazuh, or Splunk

By cyberboss