LogHawk Project Report

**Author:** Linta Susan George  
**Program:** Cybersecurity Bootcamp @ Lighthouse Labs  
**Date:** July 2025

# 📁 TABLE OF CONTENTS

* [Introduction](https://chatgpt.com/c/68746e7e-31c0-8010-ad39-51aa19579da5#1-introduction) ---------------------------------------------------------------- 3
* [Solutions Section](https://chatgpt.com/c/68746e7e-31c0-8010-ad39-51aa19579da5#2-solutions-section) ---------------------------------------------------------------- 3
* [Potential Iterations](https://chatgpt.com/c/68746e7e-31c0-8010-ad39-51aa19579da5#3-potential-iterations) ---------------------------------------------------------------- 5
* [Conclusion](https://chatgpt.com/c/68746e7e-31c0-8010-ad39-51aa19579da5#4-conclusion) ---------------------------------------------------------------- 5
* [References](https://chatgpt.com/c/68746e7e-31c0-8010-ad39-51aa19579da5#5-references) ---------------------------------------------------------------- 5

## INTRODUCTION

Log analysis plays a crucial role in cybersecurity by helping detect Indicators of Compromise (IoCs), respond to threats early, and secure critical systems. This project introduces **LogHawk**, a lightweight log monitoring tool built using Python and Bash, designed to automate the detection of common suspicious activities in logs such as brute-force attacks, system failures, and unauthorized script executions. Log monitoring is critical in detecting early Indicators of Compromise (IoCs) such as failed SSH attempts, which can indicate brute-force attacks (MITRE ATT&CK, 2024).

The motivation for this tool stems from the need to equip security teams with a simple, customizable, and open-source log scanning solution that can scale with diverse infrastructure and be easily scheduled and extended.

### SOLUTIONS SECTION

**Overview**

LogHawk includes two scripts:

* loghawk.py: a Python-based flexible log parser
* loghawk.sh: a Bash-based quick filter scanner

These scripts analyze logs from sources like auth.log, access.log, and system.log.

**🐍 Python Script: loghawk.py**

**Functions:**

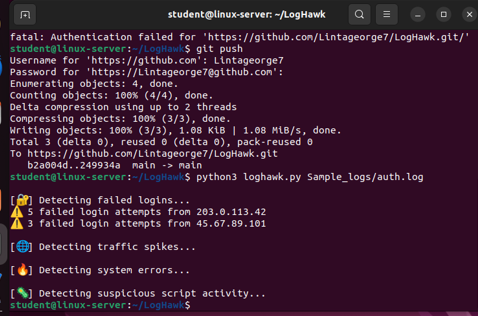
* Detect multiple failed SSH login attempts (potential brute-force)
* Identify critical system errors (ERROR, CRITICAL)
* Flag cron jobs running suspicious commands like wget, curl

**Usage:**

python3 loghawk.py sample\_logs/auth.log

**Sample Output:**

[🦠] Suspicious cron command detected: wget payload.sh



The script’s failed login detection aligns with MITRE ATT&CK Technique T1110 for Brute Force (MITRE, 2024).

**💻 Bash Script: loghawk.sh**

**Functions:**

* Uses grep to scan for failed logins, system errors, and cron activity

**Usage:**

./loghawk.sh sample\_logs/system.log

Sample Output:  
  
A computer screen shot of a program

AI-generated content may be incorrect.

The script output indicates a potential privilege escalation event, where user3 initiated a root shell using sudo (COMMAND=/bin/bash). Such activity is often scrutinized in cybersecurity monitoring due to its potential use in post-exploitation or lateral movement scenarios (MITRE, n.d.). Monitoring sudo commands, especially those invoking interactive shells, is a common security best practice (NIST, 2020).

🛡️ Security implication: This could indicate privilege escalation, especially if user3 shouldn't be running interactive root shells.

**⏲ Automation Setup**

**Linux Cron Job:**

\*/10 \* \* \* \* /home/student/LogHawk/loghawk.sh /var/log/auth.log

**Windows Task Scheduler:**

* Program: python
* Arguments: C:\path\to\loghawk.py C:\logs\auth.log
* Trigger: Every 10 minutes

#### POTENTIAL ITERATIONS

To enhance LogHawk:

* 📧 Add email alerting and threshold-based escalation
* 📊 Support JSON/CSV output for integration with SIEM tools
* 💻 Create a simple web dashboard for visualization
* 🔗 Integrate with ELK stack, Splunk, or Syslog
* 📊 Add live monitoring support with socket or file watch

#### Conclusion

LogHawk is a foundational log monitoring tool that demonstrates core cybersecurity skills such as scripting, log parsing, automation, and IoC awareness. It provides actionable insights for system defenders and can serve as a building block for more advanced threat detection systems. Its open-source nature encourages collaboration and continuous improvement from the security community.

# REFERENCES

* MITRE ATT&CK. (2024). Technique T1110: Brute Force. <https://attack.mitre.org/techniques/T1110/>
* Red Hat. (2024). *Log files and monitoring*. <https://access.redhat.com/articles/log-files>
* NIST. (2020). *Guide to Computer Security Log Management*. <https://csrc.nist.gov/publications>
* MITRE. (n.d.). T1548.003: Abuse Elevation Control Mechanism: Sudo and Sudo Caching. MITRE ATT&CK®. <https://attack.mitre.org/techniques/T1548/003/>
* National Institute of Standards and Technology (NIST). (2020). NIST SP 800-92: Guide to Computer Security Log Management. <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-92.pdf>