

Cybersecurity Mini Portfolio

Thu Thao Le

November 18, 2025

Personal Information

Name: Thu Thao Le

Email: kimtruongle6@gmail.com

GitHub: github.com/kimtruongle6-prog

LinkedIn: linkedin.com/in/kim-truong-le-8a3b15381

Motivation

I am strongly motivated to study Cybersecurity in a structured and professional environment. I have been self-learning fundamental concepts such as threat analysis, security monitoring, incident investigation and Python automation. My goal is to develop a solid technical foundation and contribute to security-related tasks with analytical thinking and discipline.

Threat Analysis Mini-Report

Summary

This report provides a simplified analysis of a simulated phishing-based intrusion scenario. It includes Indicators of Compromise (IoCs), attacker behavior mapping, and recommendations.

Key Indicators of Compromise (IoCs)

- Malicious domain: `secure-login-verification.net`
- Suspicious sender: `support@security-mail.help`
- URL path: `/update/auth/login.php`
- Attacker IP: `185.242.56.21`

Observed Techniques

- Phishing link used to harvest credentials
- HTTPS communication to evade detection
- Data collection from the user's device

Potential Impact

- Credential theft
- Unauthorized access
- Possible lateral movement

Recommendations

- Block attacker domain and IP
- Enforce multi-factor authentication
- Improve phishing awareness
- Monitor for unusual login patterns

Incident Log Analysis (Simulation)

Scenario

A user account exhibits abnormal login attempts followed by suspicious command execution. Below are simplified logs used for investigation.

Suspicious Log Entries

- Multiple failed logins from IP 185.242.56.21
- Successful login at 02:13 AM
- Encoded PowerShell command executed
- Outbound connection to unknown domain

Timeline

- 02:06 — 15 failed logins
- 02:13 — Successful login
- 02:15 — Encoded PowerShell command
- 02:17 — Outbound traffic triggered

Conclusion

The logs suggest unauthorized account access and follow-up activity using encoded commands. Immediate password reset, account lockout and further forensic review are recommended.

Python Mini-Project: IOC Checker

Goal

A small Python script that checks whether log entries contain known Indicators of Compromise.

Code Snippet

```
ioc_list = [
    "secure-login-verification.net",
    "185.242.56.21"
]

def check_ioc(entry):
    entry_lower = entry.lower()
    for ioc in ioc_list:
        if ioc in entry_lower:
            return True
    return False

log_entries = [
    "User_login_from_185.242.56.21",
    "Connection_to_example.com"
]

for log in log_entries:
    if check_ioc(log):
        print("[ALERT] IOC detected:", log)
    else:
        print("[OK] No match:", log)
```

Outcome

This project demonstrates basic automation skills for analyzing logs and identifying potential threats.

Closing Note

Thank you for reviewing my portfolio. I am eager to continue learning and building strong technical foundations in Cybersecurity.