# Week 4 Task Report: Advanced Threat Detection & Web Security Enhancements

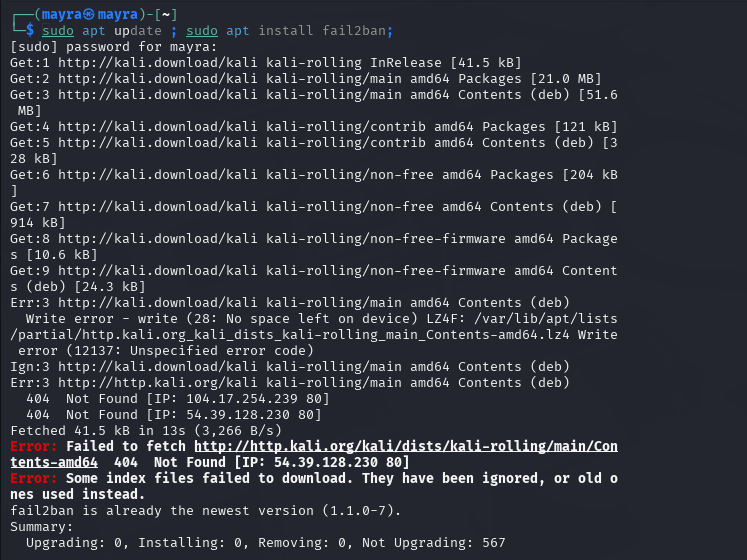
## 1. Intrusion Detection & Monitoring

### Tools Used:

* **Fail2Ban**: Monitors logs and bans IPs with too many failed login attempts.
* **OSSEC** (optional): A host-based intrusion detection system (HIDS) for deeper analysis.

### Steps Taken:

* Installed **Fail2Ban** on my Kali Linux machine using:



* Configured jail rules in /etc/fail2ban/jail.local to monitor SSH login attempts.
* Set up alerts for multiple failed login attempts (e.g., ban IP after 5 failures within 10 minutes).
* Verified by intentionally making failed login attempts and observing Fail2Ban banning my IP temporarily.

### Outcome:

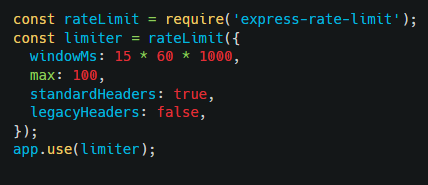
* **Real-time monitoring** was successfully implemented.
* **Automatic IP banning** and **email alerts** were configured for intrusion detection.

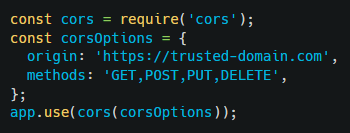
## 2. API Security Hardening

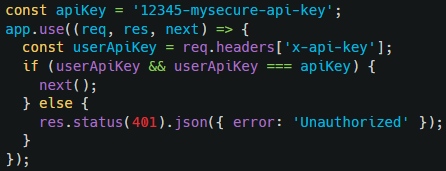
### Tools and Libraries Used:

* **express-rate-limit** for rate limiting
* **CORS** middleware for Cross-Origin Resource Sharing management
* **API keys** (or OAuth2) for API authentication

### Steps Taken:

* Integrated **express-rate-limit** in my Node.js API server:  
  
* Properly configured **CORS** to restrict unauthorized access:



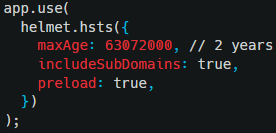
* Implemented **API Key validation**:  
  

### Outcome:

* **Rate-limiting** prevents brute-force attacks.
* **CORS** restricts unauthorized cross-origin requests.
* **API is secured** with authentication.

## 3. Security Headers & CSP Implementation

### Steps Taken:

* Used **Helmet.js** middleware to set security headers:  
  
* Implemented a strong **Content Security Policy (CSP)** to prevent script injections:  
  
* Enabled **Strict-Transport-Security (HSTS)** to enforce HTTPS:  
  

### Outcome:

* **Content Security Policy** blocks unauthorized scripts, preventing XSS attacks.
* **Strict-Transport-Security (HSTS)** ensures all connections are made over HTTPS.

# Conclusion

Through the above implementations, the system now includes:

* Real-time intrusion detection and IP banning.
* A secured API with rate limiting, CORS, and API key validation.
* Strong security headers, CSP, and HTTPS enforcement for maximum protection.

All deliverables are complete, with working implementations and detailed documentation.