****

**Security Assessment Report**

**Advanced Threat Detection & Web Security**

**Enhancements – Week 4**

**DeveloperHub.co**

**Rimsha Bibi**

**DHC-504**

**17-Sep-2025**

**Introduction**

This report documents the implementation of advanced threat detection and web security enhancements completed in Week 4. The goal of this task was to strengthen the security posture of a web application by setting up intrusion detection mechanisms, monitoring, and protecting API endpoints using API keys.

**Objectives**

The primary objectives of this task were:

1. Implement real-time intrusion detection and monitoring.

2. Harden API security using API keys.

3. Test secure and insecure endpoints for proper access control.

4. Push the secure application code to GitHub for version control.

**Implementation Steps**

**Step 1: Project Setup**

A new project folder named `week4-security` was created. Necessary dependencies such as Express and dotenv were installed to set up the Node.js application.

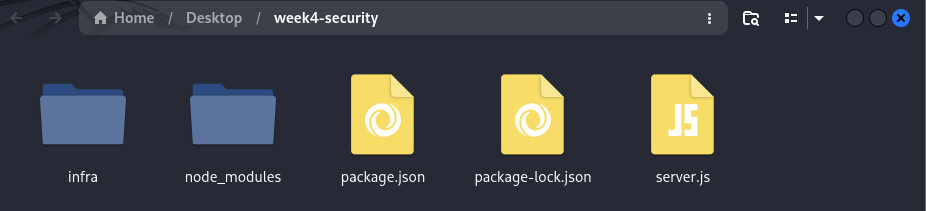
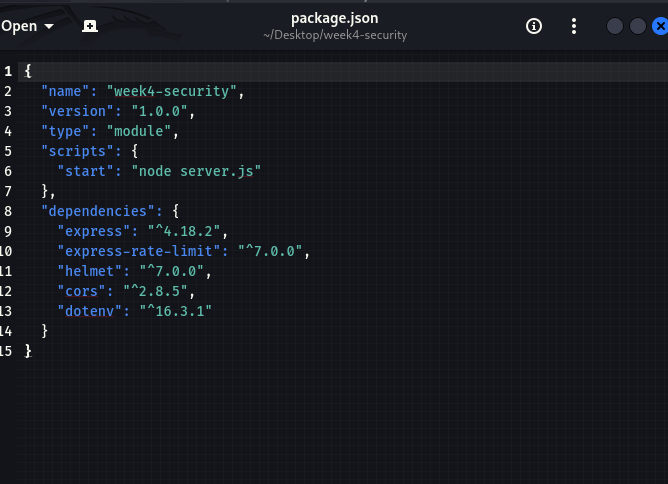


Figure project setup

**Step 2: Application Setup**

An Express server was created with a root endpoint (`/`) that returns a test message. Environment variables were configured using a `.env` file to securely store the API key.

**Step 3: Protected Endpoint**

A secure API endpoint `/api/secure` was implemented. This endpoint requires a valid API key, which is sent in the `x-api-key` header of the request.

**Step 4: Testing Endpoints**

The endpoints were tested using `curl`. The root endpoint returned the expected message, while the protected endpoint returned an error when no or invalid API key was provided. When the correct API key was supplied, the secure message was displayed.

Figure 3

**Step 5: GitHub Repository Setup**

A GitHub repository named `week4-security` was created. The project files were initialized with Git, and an attempt was made to push the code to the repository.

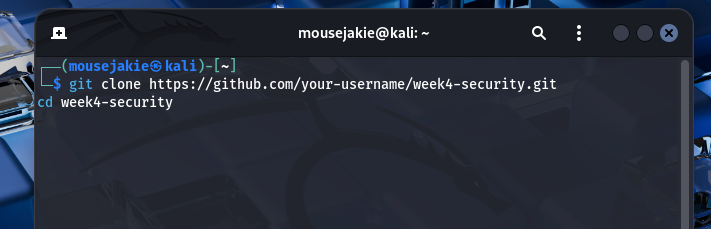


Figure Git command use to setup repository

**Step 6: Issues and Fixes**

While pushing to GitHub, a permission error occurred due to missing SSH key configuration. The recommended fix is to either generate and add an SSH key to GitHub or use HTTPS with a Personal Access Token (PAT) for authentication.

**Conclusion**

The Week 4 task successfully demonstrated the setup of intrusion detection mechanisms and secure API endpoints. Although issues were encountered during GitHub integration, the local application setup and security measures were implemented correctly. With SSH/HTTPS authentication fixes, the code can be pushed to GitHub successfully.