Project Proposal: Automated Vulnerability Monitoring and Patch Update System

Robas Ahmed Shah K21-3613
Annas Bin Saad - K21-3590

March 13 - 2024

1 Introduction

In today's cybersecurity landscape, staying informed about vulnerabilities and patch updates is crucial for maintaining the security of operating systems. This project aims to develop an automated system for monitoring firmware vulnerabilities, identifying patch updates, and providing instructions for applying patches or mitigating security exploits.

2 Objectives

- Develop a web scraping module to extract information about firmware vulnerabilities and patch updates from relevant websites and forums.
- Implement a logging mechanism to store details about identified vulnerabilities, available patches, and associated risks.
- Integrate the GPT API to generate instructions for updating patches or mitigating security exploits based on the logged data.
- Provide a user-friendly interface for interacting with the system and accessing recommendations for patch updates and risk mitigation strategies.

3 Methodology

3.1 Web Scraping Module

Identify and prioritize websites, forums, and online resources that publish information about firmware vulnerabilities and patch updates. Develop web scraping scripts using Python and libraries such as BeautifulSoup or Scrapy to extract relevant data from the identified sources.

3.2 Logging Mechanism

Design a structured log file format to store information about identified vulnerabilities and available patches. Implement logging functionality within the web scraping module to record details about each vulnerability, including its description, severity, affected firmware versions, and patch availability.

3.3 Integration with GPT API

Integrate the GPT API into the system to generate natural language instructions and recommendations based on the logged data. Develop logic to interpret the logged information and formulate queries to the GPT API for generating guidance on patch updates or security exploit mitigation.

3.4 User Interface

Develop a user interface for interacting with the system, allowing users to initiate vulnerability scans, view logged data, and access patch update recommendations. Design a user-friendly dashboard to display information about identified vulnerabilities, available patches, risk assessments, and recommended actions.

4 Expected Deliverables

- Web scraping module for collecting firmware vulnerability information from online sources.
- Logging mechanism to store details about identified vulnerabilities, available patches, and associated risks in a structured log file.
- Integration with the GPT API to generate instructions and recommendations for patch updates and security exploit mitigation.
- User interface for interacting with the system and accessing vulnerability data, patch update recommendations, and risk assessments.

5 Conclusion

The proposed project aims to develop an automated system for monitoring firmware vulnerabilities, identifying patch updates, and providing instructions for patching or mitigating security exploits. By leveraging web scraping techniques, logging mechanisms, and the GPT API, the system will empower users to stay informed about security risks and take proactive measures to protect their operating systems from potential threats.