

Text

Description automatically generatedText

Description automatically generatedText

Description automatically generated

FINAL PROJECT

CVE Reports

TABLE OF CONTENTS

[1. Objectives 2](#_Toc145495512)

[1.1. Simplified Requirements 2](#_Toc145495513)

[1.1.1. Minimum Requirements 3](#_Toc145495514)

[1.1.2. Bonus Requirements 3](#_Toc145495515)

[1.2. Deadline 3](#_Toc145495516)

[1.3. Submission Method 3](#_Toc145495517)

[1.4. What to Include in your Submission 4](#_Toc145495518)

[1.5. Peer Review 5](#_Toc145495519)

[1.5.1. Importance of Peer Reviews 5](#_Toc145495520)

[1.5.2. Ethical Conduct 5](#_Toc145495521)

[1.6. Evaluation 6](#_Toc145495522)

# Objectives

The primary aim of this group project is to develop a Python-based tool that automates the gathering and analysis of information about a given security vulnerability, identified by its CVE ID (Common Vulnerabilities and Exposures). The objective is to assist the cybersecurity team in Azerbaijan to expedite their understanding and mitigation of security risks.

**What is CVE?**

CVE (Common Vulnerabilities and Exposures) is a registry that provides a comprehensive listing of publicly disclosed cybersecurity vulnerabilities. Each entry has a unique CVE ID and includes various details such as the vulnerability's severity and possible mitigations.

## Simplified Requirements

* **Input from User**: The program should prompt the user to enter a CVE ID.
* **Data Collection**: Utilize web scraping techniques to collect information about the entered CVE ID from various online sources.
  + **Example Resources**
    - NIST NVD (National Vulnerability Database):
      * e.g. <https://nvd.nist.gov/vuln/detail/CVE-2017-0144>
    - Exploit DB:
      * e.g. <https://www.exploit-db.com/search?cve=CVE-2017-0144>
    - MITRE CVE Database:
      * e.g. <https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2017-0144>
    - Vulners Database:
      * e.g. <https://vulners.com/search?query=CVE-2017-0144>
    - Vulmon Database:
      * e.g. <https://vulmon.com/vulnerabilitydetails?qid=CVE-2017-0144>
* **Data Analysis**: The program should filter and prioritize the gathered data, concentrating on key elements like severity and remedial actions.
* **Report Generation**: Generate a well-structured report using the analyzed data.

### Minimum Requirements

* Provide a summary of the vulnerability.
* Parse and include CVSS (Common Vulnerability Scoring System) analysis from the resources.
* Automatically open **up to five** top references containing technical details about the vulnerability (**ensure the links are not broken**).
* Check for the availability of an exploit script on Exploit DB and include it in the report if available.
* Use format strings to structure the report for better readability.

### Bonus Requirements

* Enable the tool to save the report to a file on the filesystem.
  + Extra points if the saved file is in .pdf, .docx, or .md formats.
* Use additional CVE databases not listed in the example resources for data collection.
* Summarize and include additional metric data such as affected operating systems, software versions, related CVEs, and threat agents involved.

## Deadline

The final project must be submitted by **21 September 2023** at **23:59**. No late submissions will be accepted unless extenuating circumstances apply and prior permission has been granted by the instructor.

## Submission Method

Please submit your project files through the LMS. Ensure that all code, assets, and documentation files are included in a zipped folder. The folder should be named using the following convention:

**[GroupName].zip**

For example:

**GroupA.zip**

## What to Include in your Submission

Your group submission should include:

* All Python **.py** files that make up the project.
  + In the source code, clearly mark sections/lines of code written by individual group members using their initials. For example, use comments to highlight which group member was responsible for that particular code using the convention **# <initials> - <comment>**, for example, **# JH – Function to scrape NVD database**
* **requirements.txt**: A file listing all the Python packages required for your project.
  + To generate this file, you can run **pip freeze > requirements.txt** in your project directory. Make sure you're in the appropriate virtual environment, if you are using one.
* Any additional assets such as images, data files, etc., if applicable.
* A README.md file containing:
  + Project name and description.
  + Dependencies
  + Instructions for running the project.
  + List of group members and their roles.
  + Any additional instructions or comments you'd like to add.
* Individual Self-Assessments:
  + Each group member must submit a separate self-assessment document (as a **.txt** file).
  + This self-assessment should be **100-200 words** and outline your own contributions to the project, what you have learned, and how you collaborated with your team.
  + Name these files using the convention **[FirstName]\_[LastName]\_SelfAssessment.txt**, for example, **James\_Hickie\_SelfAssessment.txt**.

## Peer Review

The instructor will provide access to a form that will guide team members through the process of assessing team member contributions.

* **Be Specific**: Provide clear reasons for your ratings. Rather than saying "Good job," specify what the person did well. For example, "Developed an efficient web scraping method that contributed significantly to the project."
* **Use Evidence**: Base your assessments on observable behaviors and contributions. For example, point out specific parts of the project where the individual contributed.
* **Be Balanced**: While it’s okay to recognize excellent contributions, also point out areas that may need improvement. However, criticism should be constructive and focused on actions, not the individual.
* **Be Anonymous**: If possible, keep your reviews anonymous to encourage impartiality.
* **Conflicts of Interest**: If you feel you have a conflict of interest that prevents you from reviewing a peer fairly, bring it to the attention of your instructor.

### Importance of Peer Reviews

Peer reviews are a critical component of this group project. They provide valuable insights into each team member's contributions and help in the fair distribution of grades. It's crucial that these reviews are conducted honestly and fairly.

### Ethical Conduct

* **Be Honest, Not Generous**: The purpose of the peer review is not to be generous with points, but to provide an honest and fair assessment of each team member's contributions.
* **Avoid Collusion**: Do not agree with your team members to give each other high scores without justification. This undermines the grading process and may be considered academic dishonesty. Students found to have colluded in peer reviews, or those who have not taken the process seriously, may be subject to academic penalties ranging from loss of peer review points to academic misconduct procedures.

## Evaluation

Your project will be evaluated according to the following criteria:

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Criteria** | **Weight** |
| **Group Work** |  | **70%** |
| Input from User | * Implement and validate user input for CVE ID. | 10% |
| Data Collection | * Successfully scrape data from multiple sources and data is highly relevant. | 15% |
| Data Analysis | * Analysis, prioritizing key data like severity and possible fixes. | 15% |
| Report Generation | * Detailed and well-structured report generated. | 10% |
| Minimum Requirements | * Meets all minimum requirements as specified in the project description. | 10% |
| Documentation and Commenting | * Comprehensive and clear documentation and comments throughout the code | 10% |
| **Individual Contribution** |  | **30%** |
| Peer Review | * Each group member reviews the contributions of their teammates:   + Has the member actively participated and provided valuable, constructive feedback to peers? * This score will represents the average of the peer reviews. | 10% |
| Self-Assessment | * Each student writes a brief summary (100-200 words) about their own contributions, what they learned, and how they collaborated with the team. * Detailed self-assessment that accurately reflects individual contributions to the project. | 5% |
| Individual Code Review | * Significant and valuable contribution to the code that aligns with the project objectives. | 15% |
| **BONUS** |  | **?%** |