Project: Analyzing the Vulnerability Landscape with NVD and CISA KEV Data

Secure Systems / Empirical Security

Overview

In this project, you will analyze real-world vulnerability data from the **NIST National Vulnerability Database (NVD)** enriched with the **CISA Known Exploited Vulnerabilities (KEV)** catalog. You will use the statistical and modeling techniques covered in class to draw evidence-based conclusions about:

- vulnerability severity patterns,
- how weakness types relate to severity,
- how exploited vulnerabilities differ from non-exploited vulnerabilities,
- and what contextual signals (beyond CVSS itself) are associated with severity and exploitation.

Important: Security data is messy and incomplete. Your goal is not just to compute numbers, but to interpret them carefully and communicate what the data *does and does not* support.

Data Sources

This project uses public vulnerability data from the following sources:

- **NIST National Vulnerability Database (NVD)**: CVE records including publication dates, CVSS v3 scores, CWE weakness identifiers, reference links, and affected product information.
  `https://nvd.nist.gov/vuln/data-feeds`
- **NVD CVE JSON Repository**: Official JSON feeds containing detailed CVE records used to construct the dataset.
  `https://github.com/CVEProject/cvelistV5`
- **CISA Known Exploited Vulnerabilities (KEV) Catalog**: A curated list of vulnerabilities known to be exploited in the wild.
  `https://www.cisa.gov/known-exploited-vulnerabilities-catalog`

Key Variable: Exploitation Proxy

We join NVD and KEV by CVE ID to create a binary indicator:

$$\texttt{in\_kev} = \begin{cases} 1 & \text{if CVE appears in KEV} \\ 0 & \text{otherwise} \end{cases}$$

KEV-related columns (e.g., `kev_date_added`) will be missing for CVEs not in KEV. This is expected.

Files You Will Use

- `analysis/01_student_analysis.ipynb` (You complete this.)
- `data/processed/nvd_kev_clean.csv` (Prepared dataset you analyze.)

Deliverable

Submit your completed notebook:

`analysis/01_student_analysis.ipynb`

Your notebook must include:

- all code used to answer the questions,
- all required tables and plots,
- written interpretations in the provided markdown cells.

Environment Setup (VS Code + Jupyter)

Step 1: Install prerequisites

You need:

- **VS Code**
- the VS Code extensions: **Python** and **Jupyter**
- **Conda** (Miniconda or Anaconda)

Step 2: Create the conda environment

From a terminal in the project directory, run:

```
conda env create -f environment.yml -n nvd-kev
conda activate nvd-kev
```

Step 3: Select the kernel in VS Code

What is a "kernel"? In a Jupyter notebook, the *kernel* is the Python environment that runs your code. Selecting the `nvd-kev` kernel ensures the notebook uses the packages you installed in the `nvd-kev` conda environment.

1. Open the project folder in VS Code.
2. Open `analysis/01_student_analysis.ipynb`.
3. Click the kernel selector (top-right of the notebook UI).
4. Choose the environment named `nvd-kev`.

Troubleshooting

- If you do not see `nvd-kev` as a kernel, run:

  ```
  python -m ipykernel install --user --name nvd-kev
  ```

- If VS Code cannot find conda environments, restart VS Code after creating the environment.

- **Windows users:** If `conda activate` does not work in PowerShell, run `conda init powershell` first and restart your terminal.
- If plots do not appear inline, ensure the setup cell includes `%matplotlib inline`.

Alternative Environments

VS Code with the Jupyter extension is recommended, but you may also:

- Run `jupyter lab` from a terminal in the project directory and work in your browser.
- Use **Google Colab**: upload `nvd_kev_clean.csv` to your Colab session and update `DATA_PATH` accordingly.
- Use **pip** instead of conda: `pip install pandas numpy scipy statsmodels matplotlib` in a virtual environment.

Helpful References

Key resources (especially if you are new to Jupyter notebooks):

- **Jupyter "Getting Started"**: `https://docs.jupyter.org/en/latest/start/index.html`
- **Jupyter notebooks in VS Code**: `https://code.visualstudio.com/docs/datascience/jupyter-notebooks`
- **Jupyter kernels (what they are)**: `https://docs.jupyter.org/en/latest/projects/kernels.html`
- **pandas**: `https://pandas.pydata.org/docs/getting_started/intro_tutorials/`
- **matplotlib**: `https://matplotlib.org/stable/tutorials/pyplot.html`
- **scipy.stats**: `https://docs.scipy.org/doc/scipy/reference/stats.html`
- **statsmodels formulas**: `https://www.statsmodels.org/stable/example_formulas.html`

Project Tasks

Part 1: Exploratory Data Analysis

Compute summary statistics for `cvss_base_score` overall and grouped by severity/year. Create plots to characterize the vulnerability landscape (distribution of scores, CWE category counts, KEV vs non-KEV comparisons, trends over years). Write 2–3 paragraphs interpreting your findings.

Part 2: Chi-Square Test of Independence

Research question: Is `cwe_category` associated with `cvss_severity`? Construct a contingency table, run a chi-square test, compute standardized residuals, and interpret the results.

Part 3: Exploited vs Non-Exploited

Research question: Do KEV vulnerabilities have different `cvss_base_score` distributions than non-KEV vulnerabilities? Use a Mann–Whitney U test, compute an effect size (Vargha–Delaney A), bootstrap a CI for the difference in medians, and interpret statistical vs practical significance.

Part 4: Linear Regression (Contextual Predictors)

Research question: Which *contextual* features (not simply the CVSS formula components) are associated with higher CVSS base scores? Fit and interpret a linear regression using contextual predictors such as reference counts, patch indicators, affected-surface proxies, timing, CWE category, and year. You should expect a low R^2—think about why. Discuss model fit and what remains unexplained.

Part 5: Logistic Regression (Predicting KEV Membership)

Research question: Can we predict whether a vulnerability appears in KEV based on its characteristics? Fit a baseline model using `cvss_base_score`, then extend with contextual predictors (patch/github indicators, affected-surface proxies, CWE category, year). Interpret coefficients in terms of odds ratios and compute predicted probabilities for scenarios.

Part 6: Reflection and Limitations

Discuss one place where p-values or means alone would mislead. Connect your methodology to a paper discussed in class (e.g., Klees et al. or Dekoven et al.). Identify one important limitation of NVD/KEV data that affects your conclusions.

Use of Generative AI Tools

Students are permitted to use generative AI tools (e.g., large language models) to assist with aspects of this project, including:

- Understanding error messages or unfamiliar Python syntax,
- Debugging code or exploring alternative implementations,
- Clarifying statistical concepts or library documentation,
- Brainstorming approaches to data analysis or interpretation.

However, all submitted work must reflect **your own understanding and reasoning**. In particular:

- You may not submit code or analysis that you do not understand,
- You may not outsource the core analysis or interpretation to an AI system,
- You should be able to explain what your code does and why you made specific choices.

You are responsible for the correctness and interpretation of any code you submit, regardless of whether an AI tool was used to help generate it. Be aware that we may ask questions about your analysis or code (e.g., on the midterm or in discussion) to assess understanding.

This project emphasizes **reasoned analysis, interpretation, and clarity**. Over-reliance on generative tools without understanding may negatively impact grading, particularly in categories related to insight and explanation.