How secure is your code?

Jason **Cusati**, Faisal **Adams**, Trent **Greer** & Kirubanidhi **Ramachandran**

Problem Statement and Analysis

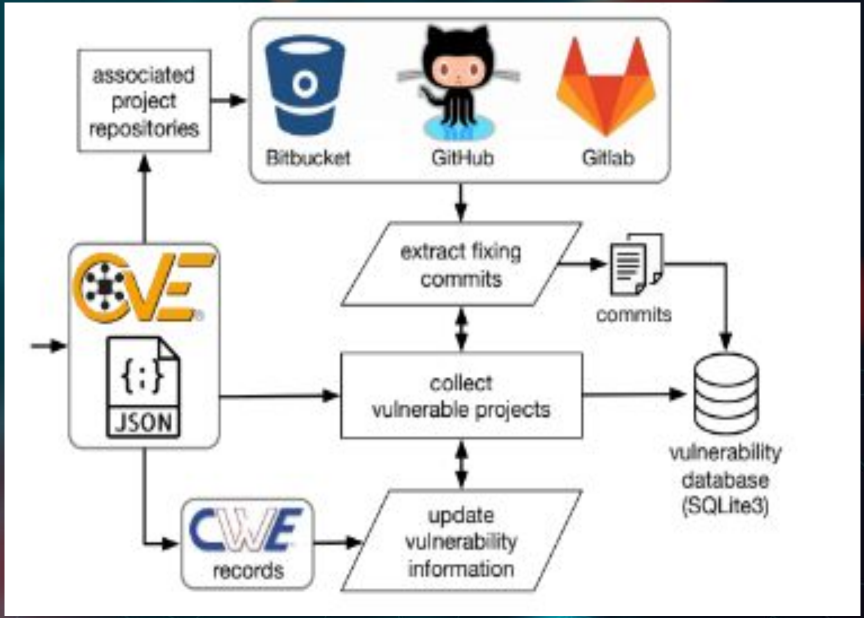
- Unsecure code can damage the economy, the government, and the people
- College Students
  - 81% of College Students perceive their own code as having low security [4]
- Chat-GPT
  - 40% of code generated by AI has security vulnerabilities
  - “Part of the problem seems to be that ChatGPT simply doesn’t assume an adversarial model of execution” [2]

Use-Case Scenarios

- A student wants to learn how to write secure code, can check their work
- Linux contributors want to maintain the security of the Linux Kernel
- A company wants to find and patch vulnerabilities in a new release
- A professor wants to demonstrate the insecurity of popular software
- Engineers can utilize this model as part of the CI/CD pipeline to keep the repo clean

Dataset Construction FLOW

- Different CVS contains CVE project
- CVE records comes as json
- All collected vulnerable projects stored in SQL Lite.
- In our project csv files are downloaded locally ,trained,modelled ,tokenized and predicted.



Data

- Common Vulnerabilities and Exposures (CVE) dataset from the U.S. National Vulnerability Database (NVD)
- Published CVEs up to 9 June 2021
- 31,160 total code entries
- 48 different programming languages
- Most common language C
- Probably correlated to most vulnerabilities (also C)

c	8632
Other	6122
php	5590
py	1564
js	1562
h	1344
java	1162
rb	1120

Models Explored and Parameters Chosen

- CodeBERT model from HuggingFace
  - Fine-tuned RoBERTa-base model
  - Utilizes both natural languages and source codes
  - Developed by Microsoft for code...
    - summarization
    - completion
    - understanding
  - All of the above makes this good for our needs

- Hyperparameters
  - constant for each model
  - 5 epochs
  - 1e^-5 learning rate
  - 8 eval/train batch size

Model Results

- 48 Models Trained
  - One for each programming language
  - Accuracy varies greatly, can exceed 75%
  - Over 65Gb of files

- Takes a few seconds to classify one file of ~100 lines
  - Extremely efficient

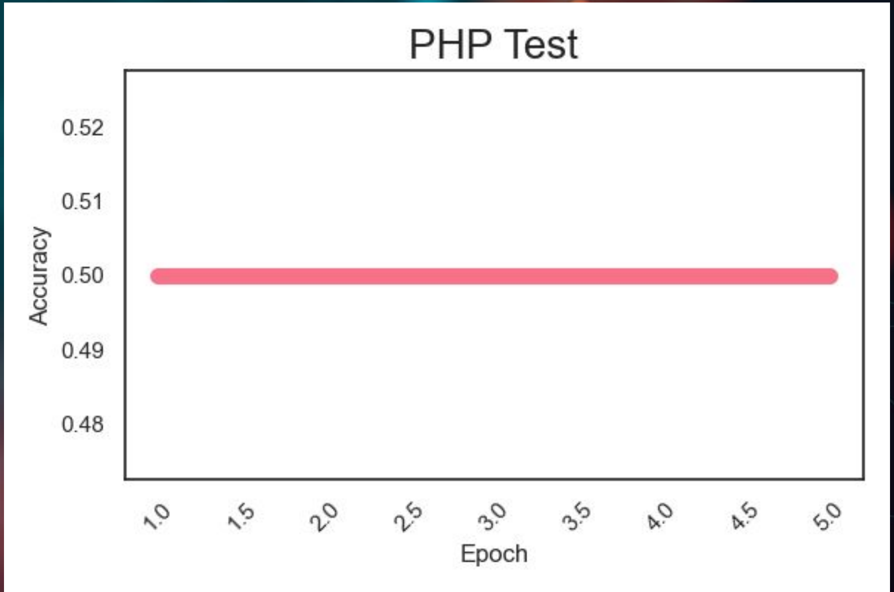
- Can classify files from command line
  - file extension is used to load correct model
  - file name and secure/vulnerable is printed to command line

Model Accuracies

- Graphs show number of epochs on the x-axis and accuracy (out of 1) on the y-axis

- Epoch with highest accuracy is what our classification program saves and uses

- Demonstrates accuracy disparity between different programming languages



Demo

Lessons Learned

- File size is large when there are multiple models

- Training time is large when training multiple models
  - Fine-tuning models creates a large computational load

- Models would perform better if training parameters were individually fine-tuned
  - This helps explain large differences in model performance

- Some models could also use more data
  - C has at least 7-8x as much data when compared to other languages
  - Contributes to disparity in model performance

- Another challenge/limitation that we have noticed is that some of the project repositories referenced in the NVD are no longer available.

Reference

1. Guru Bhandari, Amara Naseer, and Leon Moonen. 2021. CVEfixes: Automated Collection of Vulnerabilities and Their Fixes from Open-Source Software. In Proceedings of the 17th International Conference on Predictive Models and Data Analytics in Software Engineering (PROMISE '21). ACM, 10 pages. https://doi.org/10.1145/3475960.3475985
2. Feng, Z., Guo, D., Tang, D., Duan, N., Feng, X., Gong, M., Shou, L., Qin, B., Liu, T., Jiang, D. \& Zhou, M. CodeBERT: A Pre-Trained Model for Programming and Natural Languages. (2020)
3. E. Dehaerne, B. Dey, S. Halder, S. De Gendt and W. Meert, "Code Generation Using Machine Learning: A Systematic Review," in IEEE Access, vol. 10, pp. 82434-82455, 2022, doi: 10.1109/ACCESS.2022.3196347.
4. Tolga Yilmaz, Özgür Ulusoy, Understanding security vulnerabilities in student code: A case study in a non-security course, Journal of Systems and Software, Volume 185, 2022, 111150, ISSN 0164-1212, https://doi.org/10.1016/j.jss.2021.111150. (https://www.sciencedirect.com/science/article/pii/S0164121221002430)

Questions