Flipkart

GRiD 4.0

2022 Campus Challenge

Infosec Challenge

Open Source Software (OSS) Security Inspector

Team Name: H1N1

Institute Name: Vishwakarma Institute of Technology, Pune

TEAM MEMBERS DETAILS

Team Name	H1N1	
Institute Name	Vishwakarma Institute of Technology, Pune	
Team Members >	**1 (Leader)**	**2**
Name	Harsh Satpute	Niranjan Bharate
Batch	2019-2023	2019-2023

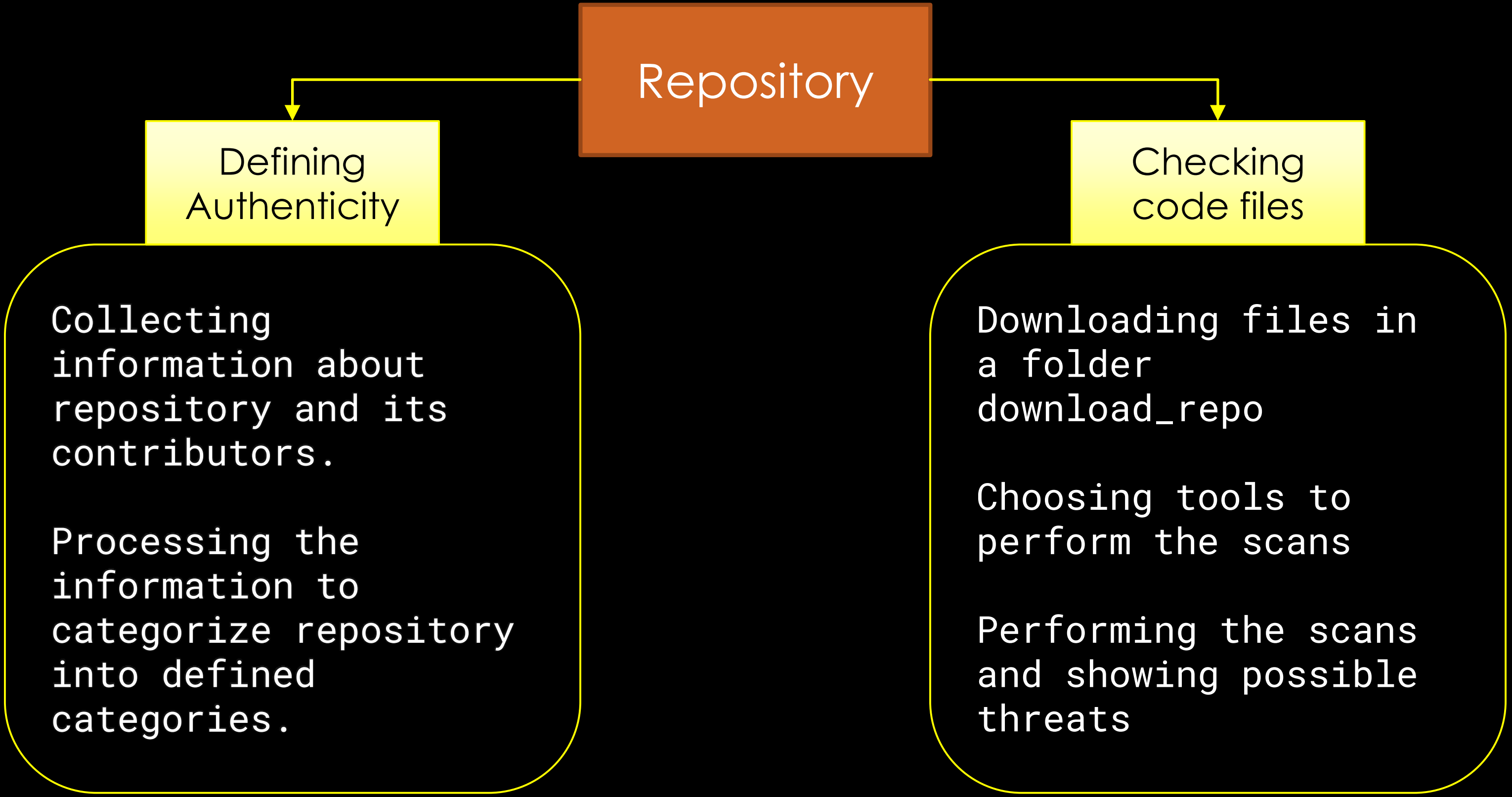
GLOSSARY

- YARA: Yet Another Recursive/Ridiculous Acronym.
- ML: Machine Learning.
- DL: Deep Learning.
- CNN: Convolutional Neural Network.

USE-CASES

- Used to validate code.
- Used to validate repository.
- Summarization of repositories rating.
- Summarization of threats in code files.

SOLUTION STATEMENT/ PROPOSED APPROACH



Repository

Defining Authenticity

Collecting information about repository and its contributors.

Processing the information to categorize repository into defined categories.

Checking code files

Downloading files in a folder download_repo

Choosing tools to perform the scans

Performing the scans and showing possible threats

COLLECTING AND PROCESSING REPOSITORY INFORMATION

(Only for GitHub)

Collected information about repository:

- Star
- Open issues
- Forks
- Contributors
- Created and last updated date

This information is fetched via GitHub API.

Processed information about contributors

- This gives us an overall idea of number of followers and public repositories of contributors.

Weighted sum of public repositories and followers of contributors

- Contributors and their number of contributions were stored in a list.
- A weighted follower and public repository count is calculated as follows

$$\frac{\sum Contributor_i(public\ repositories) \times Contribution_i}{\sum Contribution_i}$$

$$\frac{\sum Contributor_i(followers) \times Contribution_i}{\sum Contribution_i}$$

SCORE

- We wanted to calculate a score for given repository to find its authenticity.
- The parameter chosen to calculate score were weighted followers, weighted public repositories, stars and open issues.
- To calculate we need all the parameters on the same scale, this could be done if total count of each was available.
- But the total count of followers, starts and open issues on GitHub is not available.
- Number of stars is the key parameter to categorize repository.
- If the number of starts/public repositories is high, repository has a very good chance of being authentic.
- We classified them into
  - If weighted public repositories > 50 or star count > 300 : its authentic/safe.
  - Else If weighted public repositories >15 or star count >40 : its moderately risky.
  - Else its risky.

In case of GitHub, stars, open issues, forks, contributors is easily available. As well as the followers and public repositories of users are known. So, GitHub repository can be evaluated.

Whereas, in case of pypi, only the package name and it's version are available with no star/like values nor issues. And even for a particular user only his previous packages are known. So, it is quiet difficult to predict authenticity based on such limited data.

Similarly, in case of npm repositories, it's version, dependencies and collaborators can be seen with no star/like count. Only public packages of user can be seen. This again makes it quiet difficult to predict authenticity.

The values of such fields can be easily fetched from GitHub via it's API. In comparison npm and pypi don't have APIs to fetch data regarding it's packages.

DOWNLOADING REPOSITORY

- IN REPO_DOWNLOAD FOLDER, IT IS MADE SURE THAT FILES ARE DELETED BEFORE DOWNLOADING NEW REPOSITORY.
- USING GIT CLONE FUNCTION, REPOSITORY FILES ARE CLONED INTO REPO_DOWNLOAD.
- IN CASE OF PYPI AND NPM, ONLY COMMAND LINE FUNCTIONS ARE AVAILABLE TO CLONE REPOSITORIES. SO WE SUGGEST IN THIS CASE PLEASE MAKE SURE THAT THE REPOSITORIES AND CLONED MANUALLY INTO REPO_DOWNLOAD FOLDER.

YARA

- YARA is a tool aimed at (but not limited to) helping malware researchers to identify and classify malware samples.
- YARA Rule:
  - Meta: knowledge about the rule
  - Strings: All the strings that we need to search for are declared to variables.
  - Condition: A condition among the variables is declared to determine if rule is true/false.
- Rules in same category are combined together in a file.
- Rules are referred from official YARA GitHub handle.
- The rules are compiled for easy accessibility to check against repository files.
- Both separate and combined compilation is done.
- The combined compiled YARA rules file is tested against the repository in the repo_download folder.
- The code file name and the domain of suspicious part in that code file is mentioned in the output.

MACHINE LEARNING/DEEP LEARNING

- In last few days we came across research papers of malware classification. The available dataset is either in image or .csv format.
- A lot of research on android malware detection is also present.
- We decided to detect malware using ML/DL methods.
- Due to the lack of dataset in image or file format, we are not able to complete the malware detection process.
- Dataset can be compiled via multiple sources but there is limited/restricted access to most of the malware datafile collection.

  Eg. https://virusshare.com/
- Malware classification is done by ML/DL method using Convolutional Neural Network.
- The detected malware is classified into defined categories: worm, virus, trojan, backdoor, downloader, spyware, etc.

MACHINE LEARNING/DEEP LEARNING

- A DATASET OF IMAGES AS BENIGN AND MALICIOUS IS AVAILABLE ON KAGGLE. HTTPS://WWW.KAGGLE.COM/DATASETS/MATTHEWFIELDS/MALWARE-AS-IMAGES
- IT CONSISTS OF LANCZOS AND NEAREST IMAGES IN 120, 300, 600 AND 1200 DPI IN BOTH BENIGN AND MALICIOUS IMAGES.
- WE TOOK 120 AND 300 DPI IMAGES OF LANCZOS. SO, OUR DATA CONTAINED 122 BENIGN AND 151 MALICIOUS IMAGES.
- AN APPROACH OF DEEP LEARNING IS EXPERIMENTED WITH. THREE DEEP LEARNING MODELS RESNET_50, VGG_19 AND MOBILE_NET ARE TRIED AND AN ACCURACY OF 42.03%, 42.03% AND 39.13%.
- THE ACCURACIES MAY BE VERY LESS DUE TO LIMITED NUMBER OF DATASET AND IN PRE-PROCESSED FORMAT.

LIMITATIONS

- Pypi and npm packages don’t have any stars/like, downloads and followers to define the authenticity of package.
- Pypi and npm only support command line download. So they need to be downloaded manually to specified folder.
- Score to define the authenticity of repository cannot be calculated effectively as the parameters are not on similar scale and data required to bring them on similar scale is not available.
- Limited availability of dataset for malware detection and classification.

FUTURE SCOPE

- THE DATA FOR NPM AND PYPI CAN BE FETCHED USING WEB SCRAPER, BUT STILL IT WOULD BE LIMITED TO PREDICT AUTHENTICITY.

- DEVELOP A CODE TO DOWNLOAD PACKAGES FROM PYPI, NPM AND VARIOUS OTHER REPOSITORIES VIA COMMAND LINE AND OTHER MEANS.

- PROVIDING STATISTICAL DATA OF REQUIRED PARAMETERS, WE CAN DEVELOP A MATHEMATICAL MODEL TO CALCULATE SCORE TO DEFINE THE AUTHENTICITY OF THE REPOSITORY.

- BETTER DATASET AND ML/DL MODELS CAN BE USED FOR MALWARE DETECTION OF MALICIOUS ACTIVITY IN THE REPOSITORY.

THANK YOU!