FATİH SULTAN MEHMET VAKIF UNIVERSITY

Computer Engineering

Virus Detection System

Supervisor: Assist. Prof. Dr. Şaban SAHMOUD

Nagkichan MOUSTAFA IMPRAM

January 2025

Virus Detection System

01

ABOUT

Importance and Purpose of This Project

02

GOALS

What are the objectives of the project?

03

METHODS

Methods used in the project

04

TEST RESULTS

Project test outputs

05

RESULTS

Conclusion and Evaluation

06

RESOURCES

January 2025

About

The Virus Detection System Project is a customized antivirus program designed to run on the Windows operating system. It identifies security vulnerabilities and abnormal activities on the computer, informing users accordingly.

Malicious software installed on a computer can embed itself into system files, consume resources, and lead to personal privacy breaches. The goal of this project is to enhance computer security and minimize risks.

Project Goals

Early detection of malicious software such as Trojans or Worms.

Detection of anomalies by monitoring Scheduled Tasks.

Identification of suspicious programs by examining Startup Programs.

Detection of untrusted services by checking system services.

Enabling manual inspection of both suspicious and non-suspicious files by the user.

Enhancing detection methods using Machine Learning.

Methods 1/5

Various libraries and technologies have been used to detect malware and identify security vulnerabilities on the Windows operating system. C# as the programming language and Visual Studio as the development environment have been used.

Methods 2/5

APIs used to analyze the suspicious status of the obtained programs and tasks:

Hybrid Analysis API is a security service that performs dynamic analysis of malware, thoroughly examining the runtime behavior of files and URLs. In the project, this API is used for two different processes:

- Startup Programs
- Manual File Scanning

IBM X-Force Exchange API is used to check whether a task is trustworthy. This process is carried out using the method XForceExchangeService.IsKnown Malicious. The API queries the malware database using the file's hash value to determine whether the file is malicious.

OpenAI GPT API is used to retrieve detailed information about selected files.

Fast API is used for making predictions with the machine learning model. A POST method is used to send an input to the API, which includes the PE values of the relevant file. The trained model's results are retrieved using the GET method. Files predicted to be infected return a '1', and those not predicted to be infected return a '0'.

Methods 3/5

Methods Used to Create the Dataset:

Used to monitor the internet connections of files.

A program used to analyze the behaviors of infected files.

The DeepFreeze program was used to secure the test environment.

Codes developed in C# to examine system logs.

Methods 4/5

Resources:

VirusShare

The VirusShare website contains over 90 million virus files. This site is not directly accessible; an invitation is required to register. Two different zip files were uploaded and used from this site. The first file contains the MD5 hash values of 997 virus files in total. The second file contains the hash values of virus files for .NET.

MalwareBazaar

MalwareBazaar is an open-source virus sharing platform that hosts numerous known and anonymous viruses. Over 30 viruses, from different categories and anonymous sources, have been uploaded from this platform.

Methods 5/5

Methods Used for Machine Learning:

The Python programming language was used for data cleaning and model training.

After creating the dataset, it was prepared for machine learning. Empty and duplicate values were removed to obtain cleaner data

Features that were unsuitable for training the model were discarded. The most relevant features were then selected, and the feature matrix (X) and target vector (y) were prepared.

After splitting the data into 80% training and 20% testing, the model was trained using the Random Forest (RandomForestClassifier) algorithm.

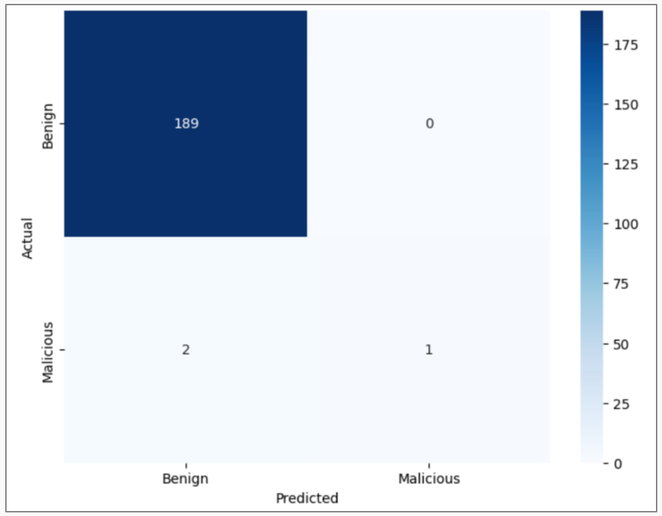
Random Forest Algorithm:

```
# Initialize the Random Forest Classifier
clf = RandomForestClassifier(
        # Set the number of trees to 100
        n_estimators=100,
        # Set the random state to 0
        random_state=0,
        # Enable the out-of-bag (OOB) score
        oob_score = True,
        # Set the maximum depth of the trees
        max_depth = 16)
# Fit the classifier to the training data
clf.fit(X_train, y_train)
```

Test Results (1/2)

Experimental Dataset Results:

Confusion Matrix

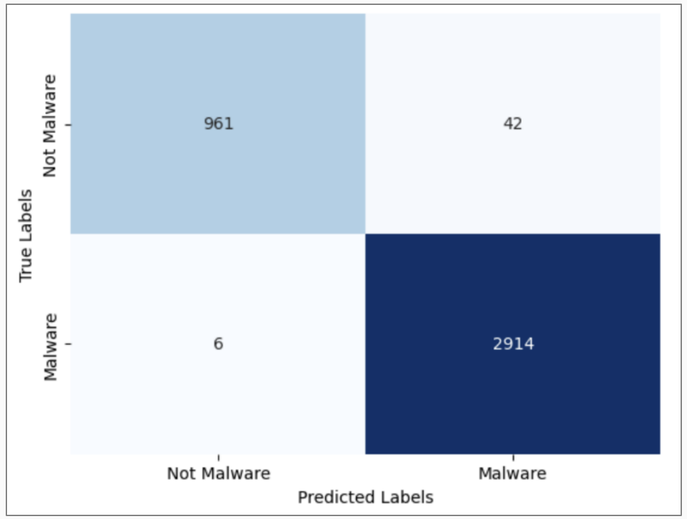


Features
CPU Usage (%)
Memory Usage (MB)
Process Name
Execution Path
Network Activities (connections and data transfer)
File Accesses
Process Runtime
PID (Process ID)
Parent Process
Labeling

Due to the amount of data used for model training and the feature selection, practical results were not successful.

Alternative Dataset Results:

Confusion MAtrix



Features
MajorSubsystemVersion
MajorLinkerVersion
SizeOfCode
SizeOfImage
SizeOfHeaders
SizeOfInitializedData
SizeOfUninitializedData
SizeOfStackReserve
SizeOfHeapReserve
NumberOfSymbols
SectionMaxChar

Due to the amount of data used for model training and the feature selection, practical results have been successful.

Test Results (2/2)

Successful Results

- All API integrations have been successfully completed. The detections have been improved to work with higher accuracy.
- Data collection and analysis have been successfully carried out, and the data has been uploaded to the database.
- A machine learning model has been created and integrated into the project.
- Errors in the user interface have been fixed.

Unsuccessful Results

- **Dataset:** The dataset used in the project has been chosen as an alternative dataset. It is not the targeted solution for this project, but it works with high accuracy.

Results

API Usage

A total of 4 APIs are used in the project. These APIs allow for more detailed and higher accuracy detections of the results. They also enhance the security of the conclusions drawn.

Detection with Machine Learning

By collecting all the processes running on the system, predictions are made using a trained machine learning model. These predictions provide high accuracy and fast results.

GPT Integration

GPT integration has been added to the project, allowing for feedback based on comments. If information is needed about a process, this method can be used in addition to API queries.

Resources

- https://www.hybrid-analysis.com/
- https://api.xforce.ibmcloud.com/doc/
- https://redcanary.com/threat-detect[on-report/techn[ques/scheduled-task/
- https://us.norton.com/blog/malware/types-of-malware
- https://visualstudio.microsoft.com/
- https://bazaar.abuse.ch/
- https://virusshare.com/
- https://www.faronics.com/en-uk/products/deep-freeze
- https://www.wireshark.org/
- https://any.run/

Teşekkürler

January 2025