|  |
| --- |
| Malware detection using CNN |
| Date of Submission – 07/03/2025 |

# Overview

## Abstract

Email remains a key attack vector for malware delivery, with traditional antivirus solutions struggling against zero-day threats. This project proposes a **CNN-based approach** for malware detection by converting binary executables into grayscale images for classification. An **Email Honeytrap** captures suspicious attachments, processes them into images, and uses a trained CNN model for real-time threat detection and quarantine. Leveraging Python, TensorFlow, and cloud services, this system enhances **malware detection accuracy** and **response speed**, demonstrating the effectiveness of deep learning in cybersecurity.

## Introduction

Email remains one of the primary attack vectors for cybercriminals, with malware-infected attachments and phishing links being common delivery methods. Traditional signature-based antivirus solutions struggle to keep up with newly emerging threats, making deep learning-based detection crucial.

**Relevance to Neural Networks and Deep Learning**

By leveraging CNNs for image-based malware classification, our system can detect structural patterns in malware binaries that traditional detection methods miss. CNNs excel in identifying hidden patterns within visual data, making them an excellent choice for malware classification.

**Problem Statement**

* Traditional malware detection methods:
* Struggle to detect zero-day malware and obfuscated binaries.
* Rely on static signatures, which can be easily bypassed.
* Cannot generalize well to new or modified malware strains.

**Research Question:**

"Can a CNN-based approach effectively classify malware binaries intercepted from email attachments, improving real-time detection and response?"

## Project Objectives

* Develop an Email Honeytrap that captures suspicious email attachments.
* Extract malicious executables and convert them into grayscale images.
* Train a CNN model to classify malware families.
* Deploy a real-time malware scanning pipeline to quarantine infected files.
* Evaluate the effectiveness of CNN-based classification against traditional detection methods.

## Methodology

**1. System Architecture**

**Intercept Emails**: Use IMAP/Gmail API to scan incoming emails for suspicious attachments.

**Download & Analyze Attachments:** If the file is an executable (.exe, .zip, .doc, .pdf), download it into a sandbox environment.

**Convert Binary to Image:** Transform malware binaries into grayscale images for CNN classification.

**Classify Malware using CNN:** Train a CNN model to classify the malware into known families.

**Threat Response System:** If malware is detected, quarantine the file and alert the user.

**2. Data Collection & Preprocessing**

**Datasets Used:**

* Malimg Dataset (PE executables converted to images)
* BIG 2015 Malware Dataset (Raw malware binaries)
* PhishTank & VirusTotal APIs (Live malware feeds)

**Preprocessing Steps:**

* Convert binaries into 2D grayscale images.
* Resize images to 64x64 pixels for CNN compatibility.
* Normalize pixel values between [0,1].

**3. CNN Architecture for Malware Classification**

Input Layer: 64×64 grayscale images

Conv2D + ReLU Activation (Feature extraction)

MaxPooling (Dimensionality reduction)

Flatten + Dense Layers (Classification)

Softmax Output Layer (Predict malware family)

**4. Training & Testing Procedures**

Split dataset into 80% training, 20% validation.

Use Adam optimizer with Categorical Crossentropy loss.

Train for 15-20 epochs on GPU for efficiency.

**Tools & Libraries Used:**

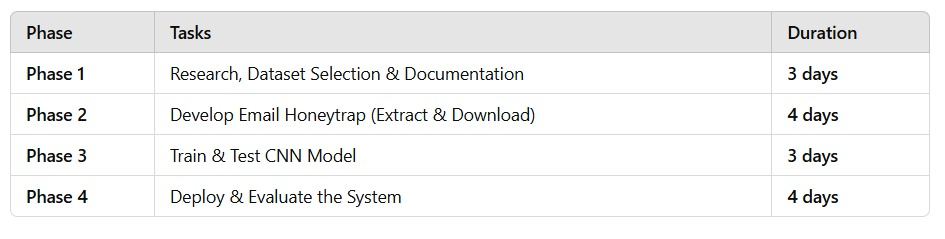
Python (TensorFlow/Keras, OpenCV, Scikit-Learn)

Email APIs (IMAP, Gmail API)

Malware Sandboxing (Cuckoo Sandbox)

Cloud Deployment (Flask/FastAPI for API, AWS Lambda for serverless execution)

## Project Plan



## Expected Outcomes

* Real-time malware detection from email attachments.
* Improved malware classification accuracy using CNNs.
* Quicker threat response via automated quarantine & alerts.
* Demonstration of deep learning's effectiveness in cybersecurity.

## Evaluation Metrics

* **Accuracy (%):** Performance of CNN in correctly classifying malware families.
* **False Positive Rate (FPR):** Avoid misclassifying clean files as malware.
* **Precision & Recall:** Ability to detect true malware samples.
* **Inference Speed:** How fast the model detects and classifies threats.

## Conclusion

This project presents a novel approach to email malware detection using CNN-based image classification. By converting malware binaries into images, we can leverage deep learning to identify structural malware patterns more effectively than traditional antivirus methods. The deployment of an Email Honeytrap will allow real-time detection and mitigation of email-based cyber threats.

# Team Details

|  |  |  |
| --- | --- | --- |
| Roll Number | Name | Section |
| CB.EN.U4CSE22015 | Guhanesh T | A |
| CB.EN.U4CSE22031 | Nalan Krishna V | A |
| CB.EN.U4CSE22036 | Prashanna R | A |
| CB.EN.U4CSE22124 | Sajeev K | B |