Topic: Malware Analysis and detection on different hosts and prediction with DL model on Wazuh

Group - Cyborg Droid
Sumit Patel (21211404), Rahul Kumar (21111069), Ashankur Tripathi (20111076),
Mandar Bapat (190475), Harishchandra patidar (21111029), Sandeep Saket (21111055),
{sumitp21, rahulkumar21, ashankkurt20, mandarb, hpatidar21, sandeeps21}@iitk.ac.in
Indian Institute of Technology Kanpur (IIT Kanpur)

1 Introduction

In today's world as cyber threats are becoming more sophisticated, real-time monitoring and security analysis are needed for fast threat detection and remediation. It has become very important to make the devices connected to internet safe and secure from the malware and viruses. The devices we use in our day to day life like laptops, phones, desktop computers etc are always connected to internet,so they are always susceptible to an attack from malware, which can infect our host systems and we can lose our important information.

So in this project we tried to use open-source tool Wazuh in order to detect and prevent from such malwares that can affect our systems.

2 Understanding the problem

- Wazuh can easily collect logs from the devices in which wazuh agent can be installed, but in this we have tried to collect logs from the devices like pen drive, android devices, router and IoT devices in which wazuh-agent cannot be installed.
- We have collected those logs and build DL model to detect the malware from executables files if it contains and classify them in different classes.
- We have shown the alerts based on the results obtained from our model in wazuh dashboard with help of custom decoders for wazuh.

3 Tools, Technologies and data used:

- Wazuh Manager

  To analyze and collect logs from different hosts.
- Wazuh Agents

  We have collected those logs and build DL model to detect the malware from executables files if it contains and classify them in different classes.

- Python Programming language

  We have shown the alerts based on the results obtained from our model in wazuh dashboard with help of custom decoders for wazuh.

- Dataset used to train model

  Requested Datasets from the author of this paper (https://arxiv.org/pdf/2103.13827.pdf)

  https://drive.google.com/file/d/1AJl5sb4iYEpPjZ4DiZfI3G_665_nO3sl/view?usp=sharing

4 Project Architecture

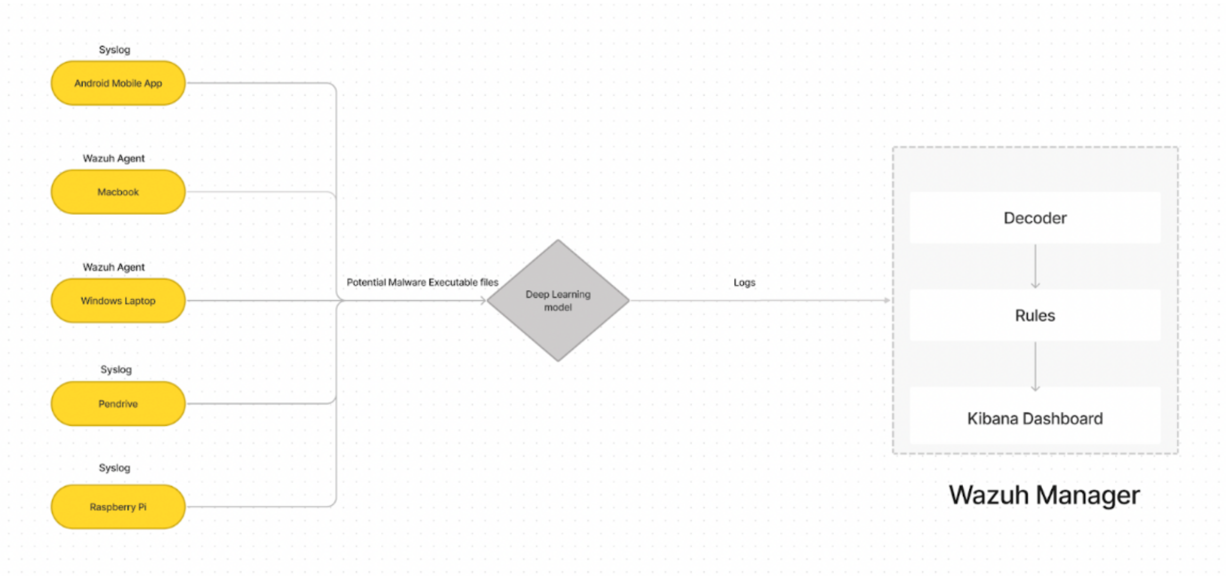


Figure 1: Project Architecture

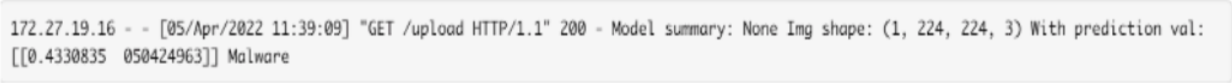
5 Collecting logs from Devices

- Linux Machine
- Macbook
- Raspberry Pi
- Pendrive
- Android Phone
- Windows machine

There are some devices like android and pendrive for which there is no wazuh agent so we collect log from them and send it to server files and then from server we append that to wazuh server. To append log from files to wauzh server we write some rules for log files like:

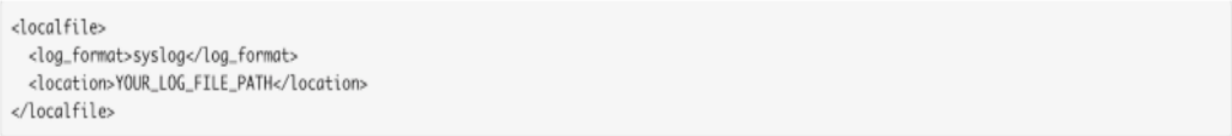
5.1 Custom RULES and DECODER for Wazuh to read logs of particular type

Logs that we are receiving from the DL model, after analysis of malware files, were in different format, which can't be read by wazuh , in order to show on the dashboard, so we created custom decoder and rules in order to read those logs and added to the respective folders in wazuh. Format of logs that we received from DL model after analysis of file:



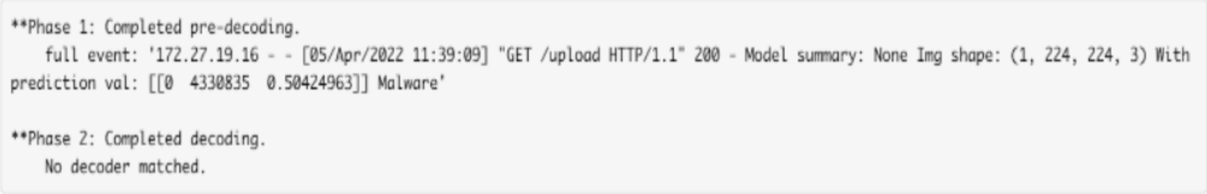
```
172.27.19.16 - - [05/Apr/2022 11:39:09] "GET /upload HTTP/1.1" 200 - Model summary: None Img shape: (1, 224, 224, 3) With prediction val:
[[0.4330835  050424963]] Malware
```

Updated configuration for wazuh in order to read logs from a custom file (dl-model.logs)



```
<localfile>
  <log_format>syslog</log_format>
  <location>YOUR_LOG_FILE_PATH</location>
</localfile>
```

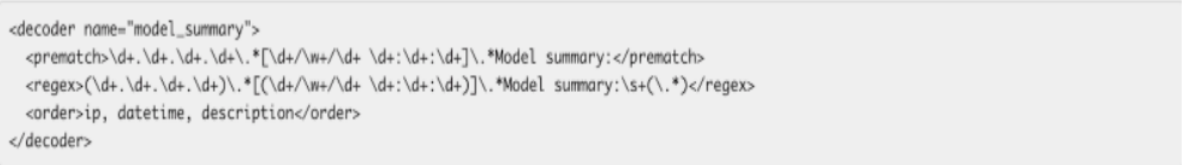
After updating configuration, we checked if there exists any decoder with following ' /var/ossec/bin/wazuh-logtest'



```
**Phase 1: Completed pre-decoding.
    full event: '172.27.19.16 - - [05/Apr/2022 11:39:09] "GET /upload HTTP/1.1" 200 - Model summary: None Img shape: (1, 224, 224, 3) With
prediction val: [[0  4330835  0.50424963]] Malware'

**Phase 2: Completed decoding.
    No decoder matched.
```

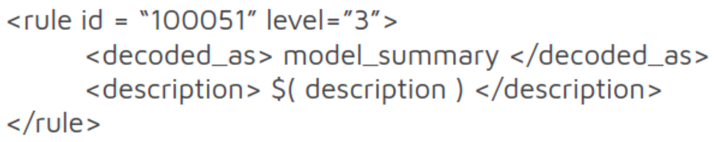
We add the following decoder to the file /var/ossec/etc/decoders/local_decoder.xml



```
<decoder name="model_summary">
  <prematch>\d+.\d+.\d+.\d+\.*[\d+/\w+/\d+ \d+:\d+:\d+]\.*Model summary:</prematch>
  <regex>(\d+.\d+.\d+.\d+)\.*[(\d+/\w+/\d+ \d+:\d+:\d+)]\.*Model summary:\s+(\.*)</regex>
  <order>ip, datetime, description</order>
</decoder>
```

6 Custom rule for Wazuh Manager

We added following rule with the id "100051" and level = 3 in /var/ossec/etc/rules/local_rules.xml



```
<rule id = "100051" level="3">
        <decoded_as> model_summary </decoded_as>
        <description> $( description ) </description>
</rule>
```

This rule will read the events from the log file and will show the respective events on the wazuh dashboard with rule id=100051

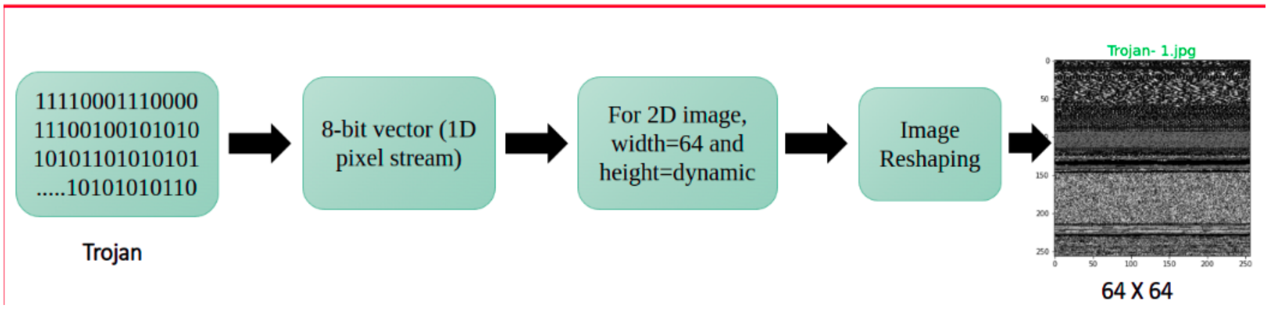
7 Approach for Malware Detection

Traditionally, malware detection has relied on pattern matching against signatures extracted from known malware. While simple and efficient, signature scanning is easily defeated by a number of well-known evasive strategies. In this project, we compare image-based deep learning (DL) models for malware analysis to a much simpler non-image based technique.

To train these DL models, we employ transfer learning, relying on models that have been pre-trained on large image datasets. Leveraging the power of such models has been shown to yield strong malware detection and classification result.

8 Conversion of Binary Executable to Gray-Scale Images

- Convert the binary executables into the sequence of 8-bit vector (1 D pixel stream)
- For converting into 2D images, the width was selected as 64 height was assigned dynamically.
- The height is calculated as Length of the pixel stream / width.
- Finally, the images were reshaped.



9 Deep Learning Model used to predict malware family

VGG-19 is a 19-layer convolutional neural network that has been pre-trained on imagenet dataset containing more than 1e6 images. This architecture has performed well in many contests, and it has been generalized to a variety of image-based problems. Here, we use the VGG19 architecture and pre-trained model as one of our two examples of transfer learning for image-based malware classification.

Our dataset consists of executables files of 20 malware families of around 1000 samples in each family. Three of these malware families, namely, Winwebsec, Zeroaccess, and Zbot, are from the Malicia dataset, while the remaining 17 families are taken from the massive malware dataset.

We first converted all executable files into binary images and then train VGG19 model to predict malware belonging to one of these families.

10 Implementation

- We have deployed this DL model on wazuh server.

Hyperparameters	
Epochs	20
Batch Size	32
Loss	CrossEntropyLoss
Optimizer	Adam
Learning rate	0.0001
Num. of classes	20

Table 1: Hyperparameters

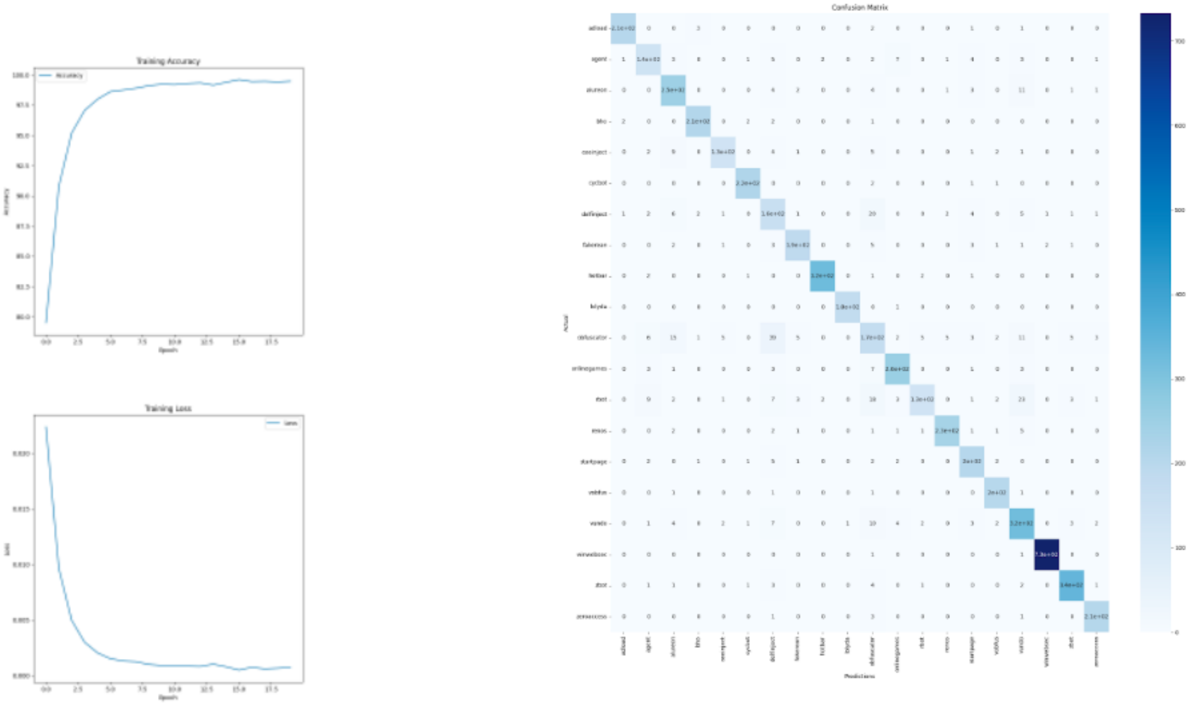


Figure 2: Training Loss and accuracy (Left) and Confusion Matrix (Right)

- We first collected all the executables files from the external devices on wazuh server and using DL model we tried to predict the malware family.
- After getting output we shown the predictions on Wazuh dashboard using simple custom decoder rules of the wazuh and added custom event to catch the logs from the file in order to show on the dashboard.

11 Results

We're getting training accuracy of 97.44% and testing accuracy of 91.11% accuracy. Table 1 shows the hyperparameters used while training the model. Figure 2 shows the training accuracy and training loss (left) and the confusion matrix(right).

12 Conclusion

- As we can see in the confusion matrix there are very less mis-classification happened while testing the model.
- Using this approach, we can monitor all external devices to detect if there's any malicious executable present in the whole system or not.
- Within few seconds model is able to detect malware if size of executable is small otherwise it depends on executable size.
- Based on alert generated on wazuh, we could take an appropriate action.

13 Limitations

One limitation is that we have considered only 20 malware families so the model will be able to predict within these 20 families.

14 Demo Video

Link for demo video - https://www.youtube.com/watch?v=vgaZpUpF3K4&feature=youtu.be

15 Future Work

- We can apply few shots learning approach when there are limited malware datasets available.
- Since there's no single way to extract images from executables, so in that case model needs to adapt the domain when there's a shift in the datasets i.e out of distributions data. We can apply Meta-Learning approach to consider multiple distributions data and make a robust model that works on all distributions data.
- We can also design a few joint BERT models for cybersecurity named entity recognition of the logs where task is to identify entities with specific meanings in the text, including names of malware, locations, organizations, specific terms related with cybersecurity, etc.