Malware Detection with Malware Images using Deep Learning Techniques

Ke He

Dong-seong Kim
Visualize a malware file as an image and use CNN to classify the image
Recurrent Neural Network in MDS

- Most frequently used deep learning technique in malware detection
- Works very well in text classification
- Programs can be thought of as text with API calls instead of words.
- Can be bypassed by adding redundant API calls which tricks the RNN\(^1\)
- RNN remembers the language environment

Convolutional Neural Network in MDS

- Often requires static/dynamic analysis to extract features.
  - doesn’t take advantage of CNN’s innate ability to extract high level features
  - We want end-to-end classification.

- Redundant API are equivalent to shifts/distortion of features in image, which a CNN is designed to recognise.

- Need to solve two major problems
  - Numerically represent malware files
  - Deal with variable size input
System Overview

**Preprocessor**
- converts binary files to grayscale/RGB images

**Classifier**
- trains the CNN with malware image and apply spatial pyramid pooling (SPP)

**Evaluator**
- Evaluates the performance of trained model
Preprocessor

- Converts each byte (0~255) to a pixel (0~255) for grayscale images
- Converts a group of 3 bytes to a pixel for RGB images
- Malware belonging to the same family exhibits similar texture\(^1\)
- All images have fixed width of 1920

---

Classifier

- Use resnet50\(^1\) or a plain network with 3 layers
- To deal with variable size inputs, spatial pyramid pooling\(^2\) is applied
- With SPP, all files are resized to (256, 256), (256, 151), (256, 433) representing the 3 main sizes of malware for training.
- Without SPP, all samples are resized to a fixed size (256,256) using bilinear interpolation

Evaluator

- Classifies with given threshold.
- Calculates the confusion matrix.
- In practice, some files are extremely large, thus we use the “divide and conquer” method.
Summary of Experiment

- Dataset Andro-dumpsys\(^1\): Android platform, 906 malware and 1776 benign file.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Level 1</th>
<th>Level 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preprocessing</td>
<td>Greyscale</td>
<td>RGB</td>
</tr>
<tr>
<td>Classifier</td>
<td>3 layer Plain</td>
<td>Resnet50</td>
</tr>
<tr>
<td>SPP</td>
<td>No SPP</td>
<td>SPP with bin size (1, 2, 4)</td>
</tr>
</tbody>
</table>

Results
Results

RoC Curves of plain model with grayscale input and no SPP on adversary inputs

RoC Curves of resnet model with grayscale input and no SPP on adversary inputs
Findings

- **SPP with divide and conquer works poorly**
  - Features are being splitted, which implies unlike traditional image classification where features are focused at one spot, malware images have features scattered around the image.
  - Images are varying too much in its size and aspect ratio that SPP cannot handle.
- **Greyscale is more resistant to redundant API calls**
  - Redundant APIs shifts subsequent pixels in RGB, can be fixed by introducing a padding scheme
- **Resizing with bilinear interpolation shows promising results against redundant API injection.**
Future Work

- Make SPP great again
- Other types of resize/compression method
- Fine tune the network
- Add distortions in input image/adversarial training
- API level image transformation
END