An Adversarial Malware Evasion Attack Using Machine Translation

Luke Kurlandski, Yin Pan, Matthew Wright

Rochester Institutute of Technology Department of Computing Security

Introduction

- Malware is a significant problem for society
- Detection systems are vulnerable to adversarial attacks
- We propose an attack that translates malicious code such that it appears benign to a deep neural classifier
- Research is in progress, no results have been obtained

Background

Adversarial Attacks

- Adversarial attacks perturb input to be misclassified
- Developing new attacks leads to improved defenses

Adversarial Malware

- Adversarial malware evades detection to harm users
- Thankfully, malware is brittle and challenging to perturb

Code Translation

- Unsupervised machine translation can translate between natural languages without parallel corpora
- Has been adapted to translate programming languages

```java
static String reverse(char[] str){
  Stack <Character> st = new Stack<>();
  for(int i = 0; i<str.length; i++)
    st.push(str[i]);
  for(int i = 0; i<str.length; i++){
    str[i] = st.peek();
    st.pop();
  }
  return String.valueOf(str);
}
```

```python
def reverse(data):
  st = []
  for c in data :
    st.append(c)
  for i in range(len(data)):
    data[i] = st[-1]
    st.pop()
  return ''.join(data)
```

Figure 1. Unsupervised code translation Java to Python, from [2]

Methods

Identifying Code A Classifier Finds Suspicious

- Neural networks can detect malware from raw bytes
- Explainability algorithms can identify suspicious code

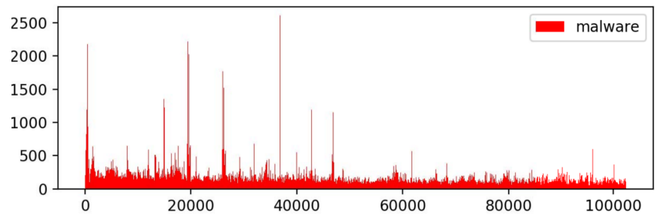

Figure 2. Large gradient activations indicate suspicious code, from [1]

Translating Suspicious Code To Appear Benign

- mal2good: a fully unsupervised malicious-looking to benign-looking assembly code translation model
- Encoder-decoder architecture trained with language modeling, denoising auto encoding, and backtranslation
- Use mal2good to replace suspicious portions of code

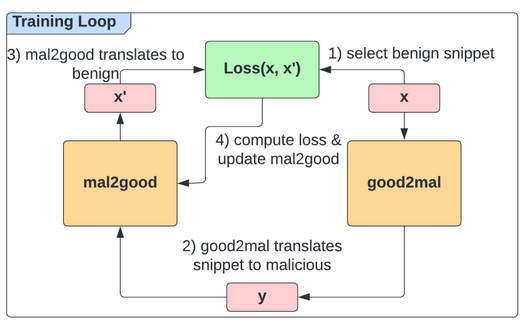

Figure 3. Backtranslation simultaneously trains forward and backward translation models (only learning forward model shown)

Experiment

- Create adversarial malware and measure evasion rate
- Test adversarial examples against additional classifiers
- Compare performance against other adversarial attacks

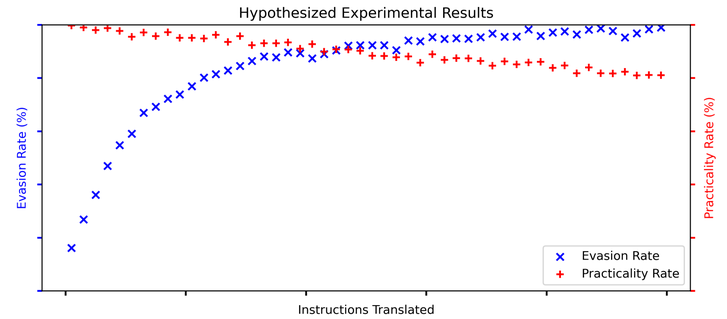

Figure 4. Translating more instructions should increase the evasion rate but decrease the percent of malware that is practical/executes properly (plot is purely speculative, no results have been obtained)

Challenges

- mal2good must produce semantically correct code
- mal2good must produce code that appears benign
- mal2good must be trained in an unsupervised manner
- Challenging to evaluate mal2good's performance

References

[1] Scott E Coull and Christopher Gardner. Activation analysis of a byte-based deep neural network for malware classification. In *2019 IEEE Security and Privacy Workshops (SPW)*, pages 21–27. IEEE, 2019.

[2] Baptiste Roziere, Jie M Zhang, Francois Charton, Mark Harman, Gabriel Synnaeve, and Guillaume Lample. Leveraging automated unit tests for unsupervised code translation. *arXiv preprint arXiv:2110.06773*, 2021.