Automated Software Vulnerability Detection with Deep Learning for Natural Language Processing

Noah Ziems (Research Scientist)
Shaoen Wu (Endowed Chair Professor)

School of Information Technology
Illinois State University
Literature Background
Traditional Methods for Detecting Security Vulnerabilities

- **Static Analysis**
  - Pre-written set of rules
  - Does not execute code
  - Prone to false positives

- **Dynamic Analysis**
  - Unit Tests
  - Written by programmer
  - Executes code
  - Vulnerabilities must be anticipated
Deep Learning Natural Language Processing for Code

- Very good at finding patterns in text
- Previous Work on Code
  - Karpathy et al. (2015)
  - Lachaux et al. (2020)
  - Rozerie et al. (2021)
  - OpenAI Codex (2021)
Software Vulnerability Detection with DL-NLP
Objectives

● General
  ○ Warn coders of security vulnerabilities in C++/Java
  ○ Specific Line

● Technical Details
  ○ State of the Art Transformer Language Model (DOBF)
  ○ Pretraining on C++/Java
    ■ Open Source Github Projects
  ○ Custom tokenizer using clang python library (C++)
  ○ function-wise evaluation
In Summary

- 76% Chance of CWE-476
- Line 3 in func1
- NULL Pointer Dereference

```
void func1(void * data)
{
    size_t dataLen = strlen((char *)data);
    void * dest = (void *)ALLOCA((dataLen + 1) * sizeof(wchar_t));
    (void)wcscpy(dest, data);
    printLine((char *)dest);
}
```
Dataset

NVD / SARD (hosted by NIST)

Input File

Text preprocessing (Customized Tool)

Formatted Texts

NLP based Deep Learning Inference Model (Transformer)

Vulnerability Type & Line

Training flow

Inference flow
Pre-Training Intuition

Source: Devlin et. al (2019)
Pre-Training & Custom Tokenization

DOBF: A Deobfuscation Pre-Training Objective for Programming Languages

Source: Roziere et. al (2021)
Architecture Diagram

[0.08, 0.19, 0.20, ..., 0.1]
[none, CWE-121, CWE-119, ..., CWE-476]

BERT

12
TRANSFORMER BLOCK

2
TRANSFORMER BLOCK

1
FEED FORWARD

1
MULTI-HEAD SELF ATTENTION

ENCODER

"int"

ENCODER

"main"

ENCODER

"}"
void func()
{
    switch(6)
    {
    case 6:
    {
        HCRYPTPROV hCryptProv;
        HCRYPTHASH hHash;
        FILE *pFile = NULL;
        char
password[PASSWORD_INPUT_SIZE];
        UCHAR savedHash[SHA1_SUM_SIZE],
calcHash[SHA1_SUM_SIZE];
        DWORD hashSize;
        char *replace;
        size_t i;
        pFile = fopen("password.txt", "r");
        if (pFile == NULL)
        {
            exit(1);
        }
        ...
    }   
}   

BERT

CWE 328: Reversible One-Way Hash

Detection Metrics(Very Good)
SARD: 93%
Draper VDISC: 98% MultiClass(5)
(https://osf.io/d45bw/)
Description Generation with Google’s T5 Architecture

- "translate English to German: That is good."
- "cola sentence: The course is jumping well."
- "sts1 sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
- "summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi..."
- "Das ist gut."
- "not acceptable"
- "3.8"
- "six people hospitalized after a storm in attala county."
Description Generation with Google’s T5 Architecture

Pre-training

Fine-tuning

President Franklin <M> born <M> January 1882.

Lily couldn't <M>. The waitress had brought the largest <M> of chocolate cake <M> seen.

Our <M> hand-picked and sun-dried <M> orchard in Georgia.

D. Roosevelt was <M> in believe her eyes <M> piece <M> she had ever peaches are <M> at our

President Franklin D. Roosevelt was born in January 1882.

When was Franklin D. Roosevelt born?

T5

1882
void func()
{
    switch(6)
    {
    case 6:
    {
        HCRYPTPROV hCryptProv;
        HCRYPTHASH hHash;
        FILE *pFile = NULL;
        char
        password[PASSWORD_INPUT_SIZE];
        UCHAR savedHash[SHA1_SUM_SIZE],
        calcHash[SHA1_SUM_SIZE];
        DWORD hashSize;
        char *replace;
        size_t i;
        pFile = fopen("password.txt", "r");
        if (pFile == NULL)
        {
            exit(1);
        }
        ...
    }
    ...
Thanks!
Questions