Automated Opcode Extraction: Implementation and Usage Report

Nisal Padukka (ID: 1353608)

February 26, 2025

1 Introduction

Malware analysis is a crucial aspect of cybersecurity, particularly in identifying threats posed by Advanced Persistent Threat (APT) groups. The **Opcode Extraction Scripts** automate the extraction of opcodes from malware samples using **Ghidra's headless analysis mode**. These opcodes are required for training a data model in the next steps of the research. This document details the tool's implementation and usage.

2 Implementation

The scripts used in this project can be found on GitHub: `https://github.com/nisalpadukka/threat-intel-ai/tree/main/opcodes`

3 Implementation Details

3.1 Overview of Components

The tool consists of two main components:

- **Wrapper Script** - Automates Ghidra's headless analysis by calling Ghidra with an external Python script.
- **Ghidra Python Script** - Extracts opcodes and saves them in CSV format (with a `.opcode` file extension).

3.2 Wrapper Script

The shell script automates the opcode extraction process by:

- Accepting a **relative malware path** as input.
- Constructing the **full path** to the malware sample.
- Running **Ghidra's headless analysis** with the Python script.

Shell Script Code:

```
#!/bin/bash

# Check if an argument is provided
if [ "$#" -ne 1 ]; then
    echo "Usage: $0 <relative_path_to_malware>"
    exit 1
fi

# Define paths
GHIDRA_PROJECT_DIR="/tmp/GhidraProject"
GHIDRA_PROJECT_NAME="MyProject"
MALWARE_BASE_DIR="/home/kali/mcti/submission_3/Executable_Malware"
SCRIPT_PATH="/home/kali/mcti/extract_opcodes.py"
OUTPUT_BASE_DIR="/home/kali/mcti/submission_3/Malware_Opcodes"

# Construct full malware file path
MALWARE_FILE="$MALWARE_BASE_DIR/$1"

# Run Ghidra Headless
./analyzeHeadless "$GHIDRA_PROJECT_DIR" "$GHIDRA_PROJECT_NAME" \
    -import "$MALWARE_FILE" -postScript "$SCRIPT_PATH" "$OUTPUT_BASE_DIR"

echo "Analysis completed for $MALWARE_FILE"
```

3.3 Ghidra Python Script

The Python script extracts opcodes from malware binaries and saves them as CSV files.
Python Script Code:

```
import os
import csv
from ghidra.app.util.headless import HeadlessScript
from ghidra.program.model.address import AddressSet

# Get script arguments
argv = getScriptArgs()
results_folder = argv[0] if argv else os.getcwd()

# Ensure results folder exists
os.makedirs(results_folder, exist_ok=True)

# Extract opcodes
program_name = currentProgram.getName()
csv_file_path = os.path.join(results_folder, program_name + ’.opcode’)

with open(csv_file_path, ’w’) as csvfile:
    csvwriter = csv.writer(csvfile)
    csvwriter.writerow([’addr’, ’opcode’])

    memory_blocks = currentProgram.getMemory().getBlocks()
    for block in memory_blocks:
        address_set = AddressSet(block.getStart(), block.getEnd())
        instructions = currentProgram.getListing().getInstructions(address_set,
        for instr in instructions:
            csvwriter.writerow([instr.getAddress().toString(), str(instr).split(
```

4 Usage Instructions

4.1 Prerequisites

- **Ghidra 11.3.1 installed** (Ensure `analyzeHeadless` is in PATH).
- **Python 3.x installed**.

4.2 Execution Steps

1. Navigate to the tool’s directory:

```
cd /home/kali/mcti/submission_3/
```

2. Run the wrapper script:

   ```
   ./analyze_malware.sh sample_malware/sample_malware.exe
   ```

3. The extracted opcodes will be saved in:

   ```
   /home/kali/mcti/submission_3/Malware_Opcodes/sample_malware/sample_malwa
   ```

5 Conclusion

These scripts automate the process of extracting opcodes from malware, allowing for streamlined data collection to train machine learning models for malware classification.

6 References

- Ghidra Installation Guide: https://github.com/NationalSecurityAgency/ghidra/releases/