CIS 600 – IoT Security and Privacy
Fall 2025

Week 10-2 - In-class Exercises: YARA Rules for Malware Detection

Total: 100 marks

Due: 2:00pm October 30, 2025

Introduction:

YARA rules are used in threat hunting to proactively search for specific patterns, strings, or byte sequences within files, network traffic, or memory to identify known or suspicious malware that may have bypassed traditional security tools. By writing and applying these rules, threat hunters can validate hypotheses, discover emerging threats, find related malware, and perform retrohunting on historical data to strengthen overall security.

In this in-class exercise, you will use YARA**,** to examine malware, identify unique indicators (strings, byte patterns, metadata), and write YARA rules to detect and classify these files. You will test, refine, and document the YARA rules that you have written for accuracy and efficiency.

Objectives:

- Analyze malware samples to extract unique indicators.
- Write effective YARA rules for malware family identification.
- Test YARA rules on multiple samples and interpret results.
- Evaluate the accuracy and performance of their YARA signatures.

Pre-tasks:

- Preconfigured virtual machine or sandbox environment with:
  - YARA installed
  - A folder containing several labeled or simulated malware samples
  - A few benign (clean) files for comparison
- Text editor (e.g., VS Code, Sublime, Notepad++)
- Command line access

What to submit:

ALL .yar files for all tasks.
A document that includes findings of each task, your observation of each task. If your rule flag any additional files beyond the malicious files that you have analyzed.

Task 1: Extract Indicators from Malware Samples

- Choose one or two malware samples from the provided dataset.
- Using tools such as `strings` or `hexdump`, identify unique patterns (e.g., embedded URLs, mutex names, or hex signatures).
- Record these indicators for later use in rule creation.

Example: The `strings` command searches for all the text strings that it can find in a file.

```
strings -n 30 ~/malware/my_sample | sort | uniq
```

It will extract all strings which are at least 30 characters long or longer from the binary while `|sort | uniq` filters those strings to remove duplicates.

Task 2: Write a custom YARA rule

Create a new `myrule1.yar` file and write a rule to detect at least TWO of the strings that stand out as unique in the malware sample.

- Include:
  - **Meta information** (author, description, date)
  - **Strings** (ASCII, wide, or hex)
  - **Condition** (e.g., number of matches or file size checks)

Note: You may include a string to search for MZ magic number and a condition to search for string at OFFSET 0 and locate based on the file hash.

Example:

```
rule Example_malware {
  meta:
    description = "Detect malware strings from xxxxxx"
    author = "Your Name Here"
    date = "10-30-2025"
    hash1 = "e128283461b14224459e966abf317c070d50aae7a531d64"

  strings:
    // Put the strings that you extracted here!
    $s1 = "string 1"
    $s2 = "string 2"
    $s3 = "string 3"
    /* ... */

  condition:
    // Add a condition using these strings to ensure that malware
    // samples get correctly identified.
    false
}
```

Task 3: Test and Refine YARA rule

Run your rule against:

- The malware sample it targets (Use the malware sample provided)
- Other malware samples (You may use any other malware file)
- Clean (benign) files
- Entire directory that you keep your malware

To execute YARA, Open a command prompt or terminal and use the `yara` executable followed by the rule file and the target you want to scan. The target can be a file, a directory, or a process ID.

```
yara <RULES_FILE> <TARGET>
```

where

- `<RULES_FILE>`: The path to your YARA rule file (e.g., `my_rules.yar`).
- `<TARGET>`: The path to the file or directory you want to scan, or the process ID.

Example:
To scan a file named `malicious.exe` with rules in `malware_detection.yar`:

```
yara malware_detection.yar malicious.exe
```

To scan an entire directory recursively:

```
yara -r malware_detection.yar /path/to/directory
```

where `-r` flag enables recursive scanning of directories.

- Adjust your rule to minimize false positives and false negatives.
- Document your observations and reasoning for each modification.

Task 4: Write a YARA rule to look for the following conditions:

A single string [md] as ASCII text, checks the first 16 bytes for the value 0x457f, specifies the file should not exceed 20,000 KB in size and that the entropy of the binary should exceed 6.5.

- Run your rule against the malware sample (Use the malware sample provided.
- Document your observations.