Simple Static Analysis

Joshua Reynolds
NMSU Reverse Engineering
Spring 2024

Helpful basic static analysis tools in Flare-VM

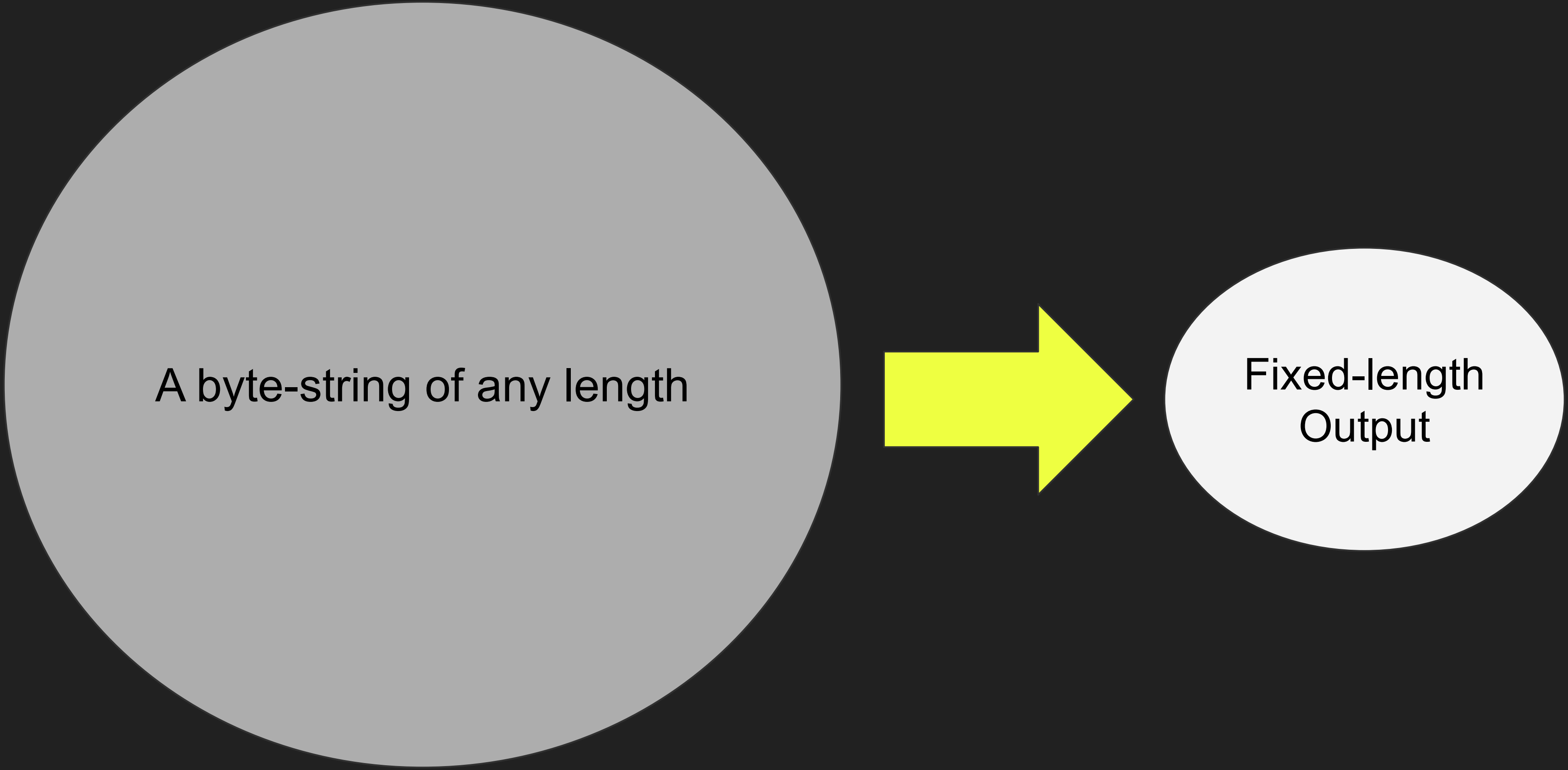
Strings (FLOSS)

PEViewer

DependencyWalker

7Zip

Hash Functions



Notable Hash Functions

MD5 - Collisions relatively easy to find

SHA-1 - Collision found by Google

SHA-256 - No known attacks, vulnerable to length extension attacks

SHA-3 - No known attacks, sponge construction prevents length extension attacks

Hashing a file in Flare-VM: CFFViewer

Open a file in CFFViewer

Find the MD5 and SHA-1 hashes

Strings (FLOSS)

Strings programs run through a file and extract any C-strings (optionally including unicode strings) of at least N characters long

You get to choose N, and the default is usually ~4

The linux utility is called strings, in flare-VM there is a similar tool called FLOSS

Run it from CMD or PS

FLOSS does more

FLOSS not only extracts static strings, but also looks for common patterns by which strings are built at runtime

It looks in 4 places:

- Static strings in the binary
- Strings built on the stack
- “Tight” strings built on the stack
- Encoded strings with a built-in decoder

Dependency Walker (Built into CFFViewer)

Shows which DLL libraries are loaded

Combined with strings (FLOSS), you can get a good idea of what functions are imported.

Libraries and their functions can be looked up on MSDN for clues about what they do.

Activity 1

Looking at the malware from this week’s assignment

Checking for external information without sharing the sample

If you share a hash, you don’t leak secrets.

If you find it in the database, someone else has seen it too

Sometimes they have already done the reverse engineering work :)

Potential References to Look up Hashes

VirusTotal

Hybrid-Analysis

All would love an upload of the actual file, but remember Op-Sec before you share. They sell/give access to samples.

Two groups of people use these services:

- malware authors who wonder if their malware has been caught
- reverse engineers who are looking for information on binaries.

Web Anonymity

If you need to look up a hash on one of these services anonymously for Op-Sec reasons, consider using TAILS

https://tails.net/

TAILS routes all traffic through TOR (The Onion Router) so these repos won't know who is talking to them (as long as you don't sign in!)

Government-level adversaries *may* still observe you by running their own TOR nodes and/or running supply chain attacks on the TAILS OS.

https://coveryourtracks.eff.org/

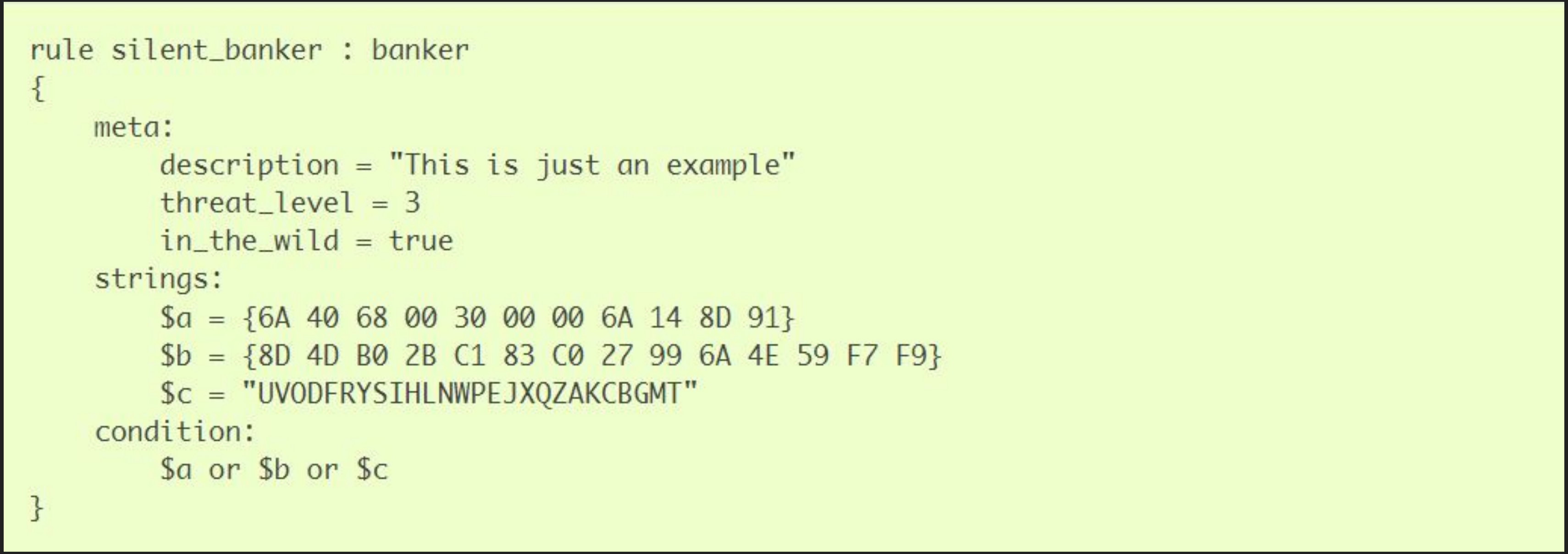
YARA - Yet Another Recursive Algorithm

A tool to scan files and compare them to known malware patterns.

https://github.com/virustotal/yara

Creating Yara Rules

https://yara.readthedocs.io/en/stable/



```
rule silent_banker : banker
{
    meta:
        description = "This is just an example"
        threat_level = 3
        in_the_wild = true
    strings:
        $a = {6A 40 68 00 30 00 00 6A 14 8D 91}
        $b = {8D 4D B0 2B C1 83 C0 27 99 6A 4E 59 F7 F9}
        $c = "UVODFRYSIHLNWPEJXQZAKCBGMT"
    condition:
        $a or $b or $c
}
```

False Positives and False Negatives in Malware Detection

False positive: benign programs are detected as malware because the rules are too broad

False negative: malware is missed because the rules weren’t general enough

Rules that are very specific may miss new versions of the same malware

Activity 2:

Creating YARA Rules from simple static analysis information