Synapse Bootcamp

Module 15

Static Malware Analysis

v0.4 - May 2024

Objectives

- Define static malware analysis
- Identify key data model elements related to static malware analysis
- Understand common pivots and queries to use
- Understand how to use relevant Power-Ups to obtain and enrich data

What is Static Malware Analysis?

- Examining the "static" properties of a file
  - Size
  - Standard hashes (MD5, SHA1, etc.)
  - MIME-specific metadata
    - PE metadata
    - Document metadata
    - Email headers
  - Strings
  - Tags (e.g., if provided by third-party)
  - Static detection data (antivirus signatures, YARA signatures)
- Contrast with "dynamic" (execution data)

Threat Intelligence and Malware

- Is this file malicious?
- What does it do / what are its capabilities?
- Have we seen it before?
  - Known malware family?
  - "Similar" samples?
- Do we know who uses it?
  - Type of activity?
  - Known threat group / groups?
- How can we detect it?
  - Unique characteristics / properties?
  - Signature-based detection?
- Can we find related samples / indicators?

Files in Synapse

- A file (`file:bytes` node) is a fundamental object in Synapse
  - Any file - malicious or not - is a `file:bytes` node
- Reports, feeds, etc. provide **hashes** that represent files
- We often model the **hash** (e.g., hash:sha1)
  - Leverage Power-Ups to obtain:
    - Information **about** the file represented by the hash
    - A **copy** of the file associated with the hash
- A `file:bytes` node is a **representation** of a file
  - Does not mean we have the **actual** file
  - Actual file (if available) stored in an Axon

Basic File Data

Data	Form
A file	`file:bytes`
A MIME type	`file:mime`
A file associated with a MIME type	`file:ismime`

The FileParser Power-Up can populate the `:mime` property for many common MIME types.

MIME-Specific File Data

MIME Type	Forms for Metadata	Example Data
PE	`file:bytes:mime:pe:*` `file:mime:pe:*`	Compile time, import hash, named export, section, resource...
message/rfc822	`inet:email:message` `inet:email:message:header`	Date, subject, from, to, body...
x509 certificate	`crypto:x509:cert`	Subject, issuer, fingerprints, validity dates...
mach-o	`file:mime:macho:*`	Section, segment, version...

RTF, MS Office, JPG, LNK, etc.	`file:mime:rtf`
	`file:mime:ms*`
	`file:mime:lnk`
	`file:mime:*`
		Application, author, last saved, imageid, title...

The FileParser Power-Up can model (or at least extract) **metadata** for many common MIME types.

MIME-Specific (Metadata) Example

file:bytes=sha256:b9f0c34f879658596a99a263c0c94d0aea6c6459bd6fcdc3276d2d4dfa48c633	
:mime	application/vnd.microsoft.portable-executable
:mime:pe:compiled	2017/12/06 14:28:31
:mime:pe:exports:libname	sc_loader.dll
:mime:pe:exports:time	2017/12/06 14:28:31
:mime:pe:imphash	d9fd8f4a0ab62a35d2c62ef2679ce79c
:mime:pe:pdbpath	c:/users/develop_mm/desktop/sc_loader/release/sc_loader.pdb
:mime:pe:richhdr	b751599bd3d9c5b67b8e1cdf012b0fa1e02b34fde18e1f0a8c4f54da463cdb14

A lot (but not all!) of PE metadata is stored directly on the `file:bytes` node.

MIME-Specific (Metadata) Example

file:mime:msdoc=bcd45caf07ec2fbd2e9862765e182b30	
:application	Microsoft Office Word
:author	Jack
:created	2017/04/26 06:33:00
:file	sha256:026e9e1cb1a9c2bc0631726cacdb208e704235666042543e766fbd4555bd6950
:file:data	{"cp:lastModifiedBy":"Windows User","cp:revision":"2","Template":"Normal.dotm","TotalTime":"2",...}
:lastsaved	2017/04/26 06:33:00
:title	Russia

Protip: In some cases, running `fileparser.parse --debug` may display additional details.

C2 Configuration Data

- Represent C2 configuration with **`it:sec:c2:config`**
  - Link to `file:bytes` node, as well as to:
    - Servers, decoys, proxies, etc
  - Capture the associated malware family name, mutex, campaign code

- Some Power-Ups will create `it:sec:c2:config` nodes for you
  - **Synapse-Fileparser**
    - Will extract and model C2 configuration data for Cobalt Strike Beacon
  - **Synapse-VirusTotal** (paid API key)
  - **Synapse-Group-IB** (paid API key)

C2 Configuration Example

it:sec:c2:config=4f962e124f54529666901e2c4955717d	
:family	cobaltstrike
:file	sha256:e075e35f74df484366f5a1497ebeb7262c16e6dad0ed6eadd18c11b0a512c7a0
:servers	(tcp://www.cybereason.xyz:443,)
:raw	<JSON blob containing the raw configuration extracted from the binary>
.created	2024/02/26 18:38:17.418

Detection Data

Detection	Form	Related Forms
Antivirus / Antimalware	`it:av:scan:result` (individual engine) `:verdict` `:multi:count:*` (summary results)	`it:av:signame`
YARA rule	`it:app:yara:match`	`it:app:yara:rule`
Generic rule	`-(matches)>` light edge	`meta:rule`

The YARA Grid Power-Up supports highly efficient scanning of files (`file:bytes`) in an Axon against a set of YARA rules stored in Synapse (along with any subsequent rule matches).

Other File Data

Description	Form
A file name	`file:base`
A file path	`file:path`
A file (`file:bytes`) seen at a specific location or with a specific name	`file:filepath`
A file on a host filesystem	`it:fs:file`
A file signed with an x509 certificate	`crypto:x509:signedfile`
A file that "contains" another file	`file:subfile` `file:archive:entry`

Common Static Analysis Tasks

Question	Workflow
What can I determine about this file?	Examine basic properties Leverage Power-Ups to enrich
Is this file malicious?	Leverage detection data Use FileParser's strings / hex viewers
Can I identify other similar files?	Pivot from file / file metadata properties to find similar files Use Power-Ups / detection data to find files detected by the same signatures If files are tagged, lift other files with the same tags

Common Tag Examples

Assessment	Tag Format (Your Assessment)	Example	Third Party
Is malicious	`#cno.mal`	`#cno.mal`	`#rep.eset.mal`
Associated with a malware family	`#cno.mal.<family>`	`#cno.mal.redtree`	`#rep.eset.industroyer`

Associated with a threat group	`#cno.threat.<group>.own`
	`#cno.threat.<group>.use`
		`#cno.threat.t872`
		`#cno.threat.t872.own`
		`#cno.threat.t872.use`
			`#rep.microsoft.nickel`

Has certain capabilities or demonstrates use of certain TTPs	`#cno.ttp.<category>.<sub>`	`#cno.ttp.pack.upx`	

You can use **triggers** in Synapse to automatically apply tags when certain conditions are met!

Static Malware Analysis - Demo

Summary

- **Static malware analysis** involves looking at static properties of a file
  - File properties
  - MIME-specific metadata
  - File name and path data
  - Detection from systems that scan files 'at rest'
- Power-Ups such as **FileParser** and **YARA grid** support static analysis
- Third-party Power-Ups may provide:
  - Ability to download files
  - File metadata
  - Detection data
  - Third-party tags