Triage for Windows Systems with Autopsy

Luca Tännler, Mathias Vetsch
26th October 2016
The Triage process

- Known Good
- Known Bad
- Investigation Bucket
Golden Image Module

Golden Image

Dirty Image

Golden Image Module

Result

Investigation Bucket
The Problem

- Suspicion a Windows computer was compromised
- What happened / changed?
- Huge set of data to be analyzed
The Golden Image Module

- What is a “Golden Image”?
  - f.Ex. Default Windows installation image
  - Often used in companies

- What is a “Dirty Image”?
  - Image based on the “Golden Image”
  - Possibly compromised / infected

- What does it?
  - Compare two images
    - “Dirty Image” vs. “Golden Image”

- What is it good for?
  - Reduce the set of data for further analysis
Procedure

- Iterate through “Golden Image”
  - Check if file exists on “Dirty Image”
  - Hash Files
  - Compare Files
- Tag Files
  - “Safe”
  - Deleted
  - Changed

Diagram:
- Dirty Image
  - Check File-Existency
    - No
      - File was deleted on Dirty Image
        - Hash File
          - Hash Identical?
            - Yes
              - File is “Safe”
            - No
              - File was changed
        - Yes
          - File is “Safe”
  - No
    - File was deleted on Dirty Image
      - Hash File
        - Hash Identical?
          - Yes
            - File is “Safe”
          - No
            - File was changed
      - No
        - File was deleted on Dirty Image
          - Hash File
            - Hash Identical?
              - Yes
                - File is “Safe”
              - No
                - File was changed
Benchmarks: Reboot

- Changes after a Reboot
- Specifications:
  - Windows 8
  - HDD: ~60GB

- Good: 240,000
- Changed: 3587
- Deleted: 3459
- New: 429
File Search / Comparison
- Independent from *filename* and *path*
- Meta Data comparison
Eliminate Known Windows Files
Eliminate Known Windows Files

- **NSRL HashSet**

  - 10.6% NIST NSRL Hits
  - 89.4% Unidentified Files
Eliminate Known Windows Files

- AuthentiCode
  - Integrity
  - Publisher Identification
  - No Quality mark
  - 2 different types

Embedded Signatures

Detached Signatures

- 00e6f1d5c22...b402505
- 15766f2be92...411ae7a
- c2d57383936...d14b523
- ee680b01dba...11cdfa5
Eliminate Known Windows Files

Embedded Signatures

File Ingest Module
1. Look for a signature in the PE Header
2. Verify the certificate
3. Check the integrity of the Binary

Tag all matches with the publisher Name

Detached Signatures

Data Source Ingest Module
1. Find all signed *.cat Files on a data source
2. Create a data structure with all signed hashes
3. Compute hashes of all files on the data source
4. Find the hashes in the data structure from step 2
The files are now grouped by Publisher
Certificate Metadata in a Content Viewer Module

The following AuthentiCode Information where found on the Image

Signature Found in:  vmusbmous.e.in
Signer:  C=US, ST=Washington, L=Redmond, O=Microsoft Corporation, OU=MOPR, CN=Microsoft Windows Hardware Compatibility Publisher
Issuer:  C=US, ST=Washington, L=Redmond, O=Microsoft Corporation, OU=Copyright (c) 2002 Microsoft Corp., CN=Microsoft Windows Hardware Compatibility PCA
Valid From:  Mon Dec 19 23:40:20 CET 2011
Valid Until:  Tue Mar 19 23:40:20 CET 2013
Eliminate Known Windows Files

- Benchmarking

![Pie charts for Signature Type and Publisher](image.png)

- Embedded Signatures: 4.81%
- Detached Signatures: 21.81%
- Files without Signature: 73.37%
- Microsoft: 95%
- 3rd Party (Drivers): 5%
TagFilter Module
The Problem

- Several tags
  - Some have same/similar meaning
  - f.Ex. “Known-Good/Bad” Tags
- Currently in Autopsy: List per tag
- Limited Filters
- No listing of all “Known Good/Bad” files
The Solution

The Tag Filter Configuration tool allows for the creation of filters with AND/OR operators, filter groups, and supports conditions like 'Contains' and 'Doesn't Contain'. It is designed to mimic the structure of SQL WHERE clauses, providing a flexible and powerful filtering mechanism.
Advanced filter options
- Change-/Create-date
- Further Meta Data

Export of results

Add Feature: Save Filter

Run modules over filtered data
VirusTotal Module
Find Known Bad Files

- Online Virus Database
- API key required
- Request are limited
Find Known Bad Files

There are more possibilities

- OpenIOC Scanner
- Cuckoo Sandbox Interface
About Autopsy

What we loved

- Open Source
- Extensible by design
- Provides a broad base

What we missed

- Documentation coverage is low
- No dynamic workflow possible
  - Not possible to run modules over set of data/file
- Content is not indexed properly
Q & A