Visual Network and File Forensics

Ankur Tyagi

@7h3rAm

Objectives

- Discuss the effectiveness of visual tooling for malware and file-format forensics (using structural analysis and visualization)
- Demo the framework that analyzes binary blobs (Pcap and PE files)
- Scan and generate reports with file's structural properties, entropy, compression ratio, minsize, file-format specific information (Yara, Shellcode, IOCs, etc.)
- Discuss clustering and classification usecases

Introduction

- Most files have a deterministic structure
- This structure can be used for identifying a candidate filetype from a pool of unknown files
- Combined with static parsing of a file and dynamic behavior captured from a sandbox, visualization of the file's structure complements analysis process

Introduction

- Rudra is a framework for analysis of network flows and PE files
- Exposes Python API for developers to integrate in their tools
- Supports plugin-based architecture
- Provides JSON reports enabling non-Python tools to leverage its analysis features

Features

- File Metadata:
  - Structural properties
    - Hashes: Unique (md5/sha) and CTPH (ssdeep)
    - Entropy, compression ratio, minsize
      - remember: entropy is indirectly proportional to compression ratio (high entropy => low compression ratio)

Features

- File Metadata:

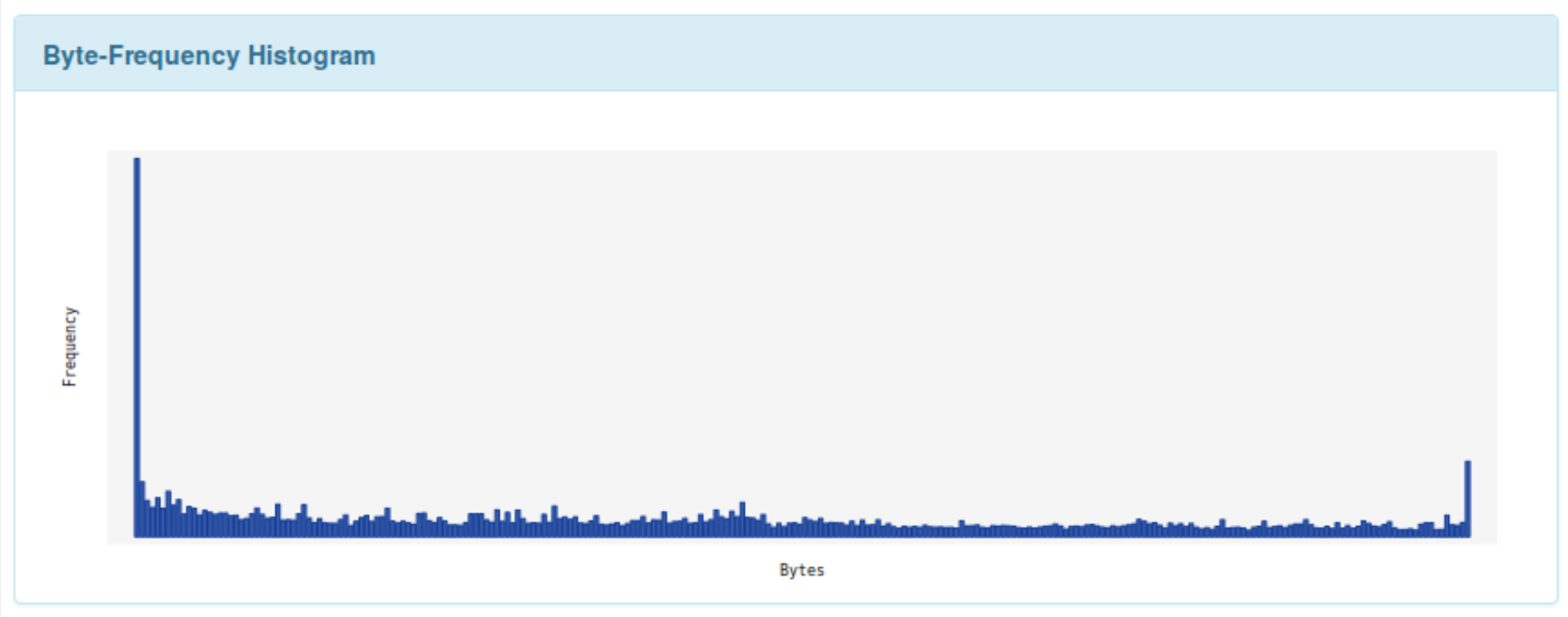
Filename	0ea4dcc92a9d9ac4ef7daff7aaaeaad5.upx.pcap
Magic	tcpdump capture file (little-endian) - version 2.4 (Ethernet, capture length 32767)
MIME Type	application/vnd.tcpdump.pcap
MD5	b2cadc0d87c9eecd5fdd3590c90895dd
SHA1	9cb2c53a9f07a747ee75347551fbdfc92af9b8e4
SHA256	2a981f7948e3926c7d545ed71e2312cb98f3ad6993be750ae86d8ee8c933ca5a
ssdeep	3072:XM8vP8QUPX2zPEYCmv2EmHAybT2bS/CJZ282N5byfBCHyB/rY8D:1LGX2zPEIoT2bS/uZ282N8fBCHydY8D
Actual Size	160.8 kB (160813 Bytes)
Min Size	152710.95
Compression Ratio	5.04%
Entropy	7.6 SUSPICIOUS

Features

- File Metadata:
  - Visualization
    - Byte-Frequency histogram
    - Raw bytes to grayscale and colored bitmaps

Features

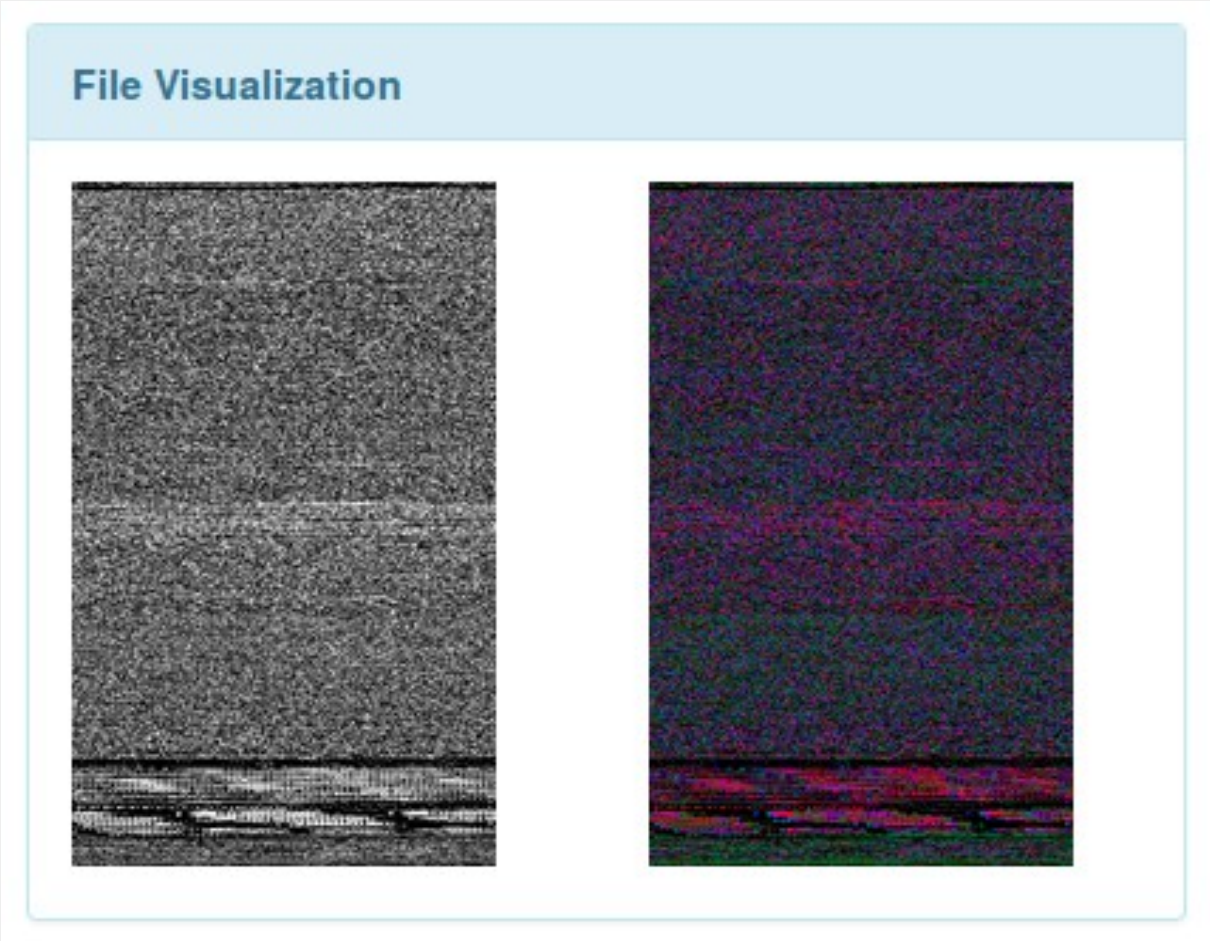
- Byte-Frequency Histogram:



Just a fancy way of snapshotting a file, ignoring byte sequence

Features

- Bytes to Bitmap:



References:
Curtis Mattoon:
http://cmattoon.com/visual-binary-analysis-python/

Aldo Cortesi:
https://corte.si/posts/binvis/announce/index.html

B. S. Manjunath:
http://sarvamblog.blogspot.com/2013/04/clustering-malware-corpus.html

Features

- File Metadata:
  - Scanning
    - Regex: detection via JSON formatted signature files
    - Fuzzy string: similarity search (ascii strings only)
    - Shellcode: x86 opcode emulation for embedded shellcode detection
    - Yara: rules based detection of document exploits and malware binaries
  - Misc
    - Embedded files identification and extraction

Features

- Pcap format specific:
  - IP defragmentation and TCP reassembly
  - Protocol identification (HTTP, FTP, SMTP, IMAP, POP3, DNS, SMB/RPC, SIP, SSDP)
  - Protocol decode (HTTP and DNS)
  - Whois and geolocation for identified hosts
  - Network traffic and file inspection
  - Visualization of protocol and file's structural properties

Features

- PE format specific:
  - Hashes: imphash and pehash
  - Extracting resources, overlays, etc.
  - Debug/PDB section parsing (RSDS/CodeView)
  - TLS parsing, strings-{ascii, unicode} extraction, anti-{debug, sandbox, vm} detection, blacklisting api imports and mutexes
  - Hash based online lookup, whitelisting using bloomfilters, etc.

Demo

Conclusion

- Visual analysis can not replace but instead complements static/dynamic analysis

- Lots of awesome tools already available. Extend them, create your own and share with the community

- Look for generic patterns, mismatch between claimed/expected vs actual content

Thanks for your attention.

Questions?

github.com/7h3rAm/rudra

github.com/7h3rAm/flowinspect