Visual Analysis of Malware Behavior Using Treemaps and Thread Graphs

Philipp Trinius, Thorsten Holz,
Jan Göbel & Felix C. Freiling

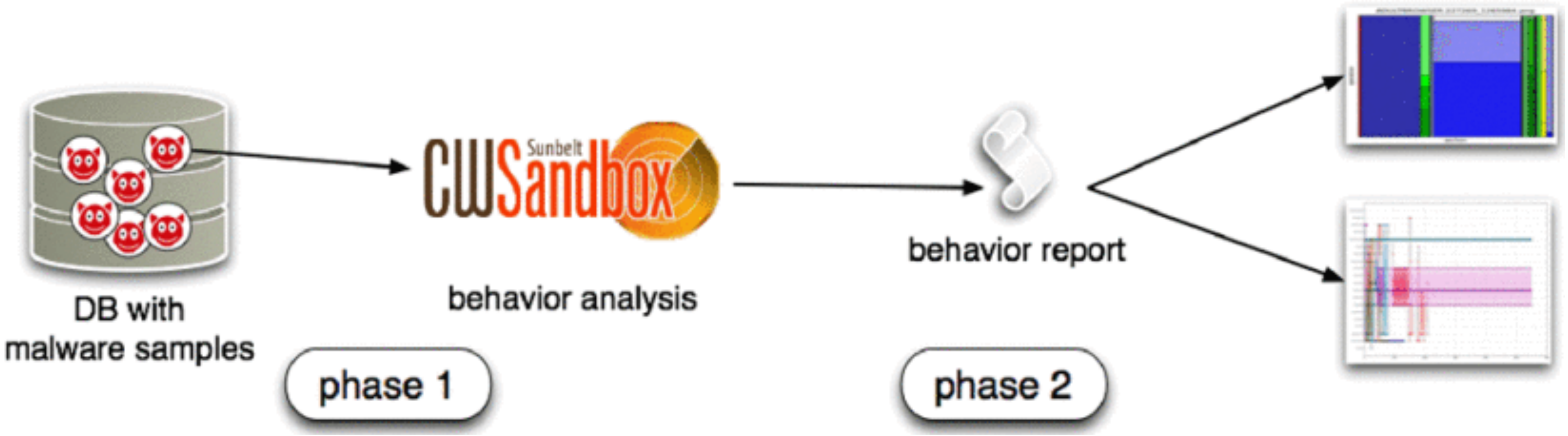
Paul Soper

CISC850
Cyber Analytics

Overview

- Dynamic analysis
- Results abstracted into small summaries
- Visualized using
  - Treemaps
    - Relative frequency of system calls
  - Thread graphs
    - Temporal behavior

System Design



Method

- Run in CWSandbox for two minutes
- Record all system-level activity to an XML file
- Perform abstractions (next slide)
- 2,500 to 4,000 reports per day are generated

Data Organization

- API calls with similar functionality grouped together in *sections*
- Arguments of each API sorted in order of decreasing relevance

Abstraction

- Abstraction levels
  1. Sections only
  2. + names of API calls in each section
  3. + information about most the most significant arguments
  4. + information about additional arguments
- Levels 2 and 3 are the most useful

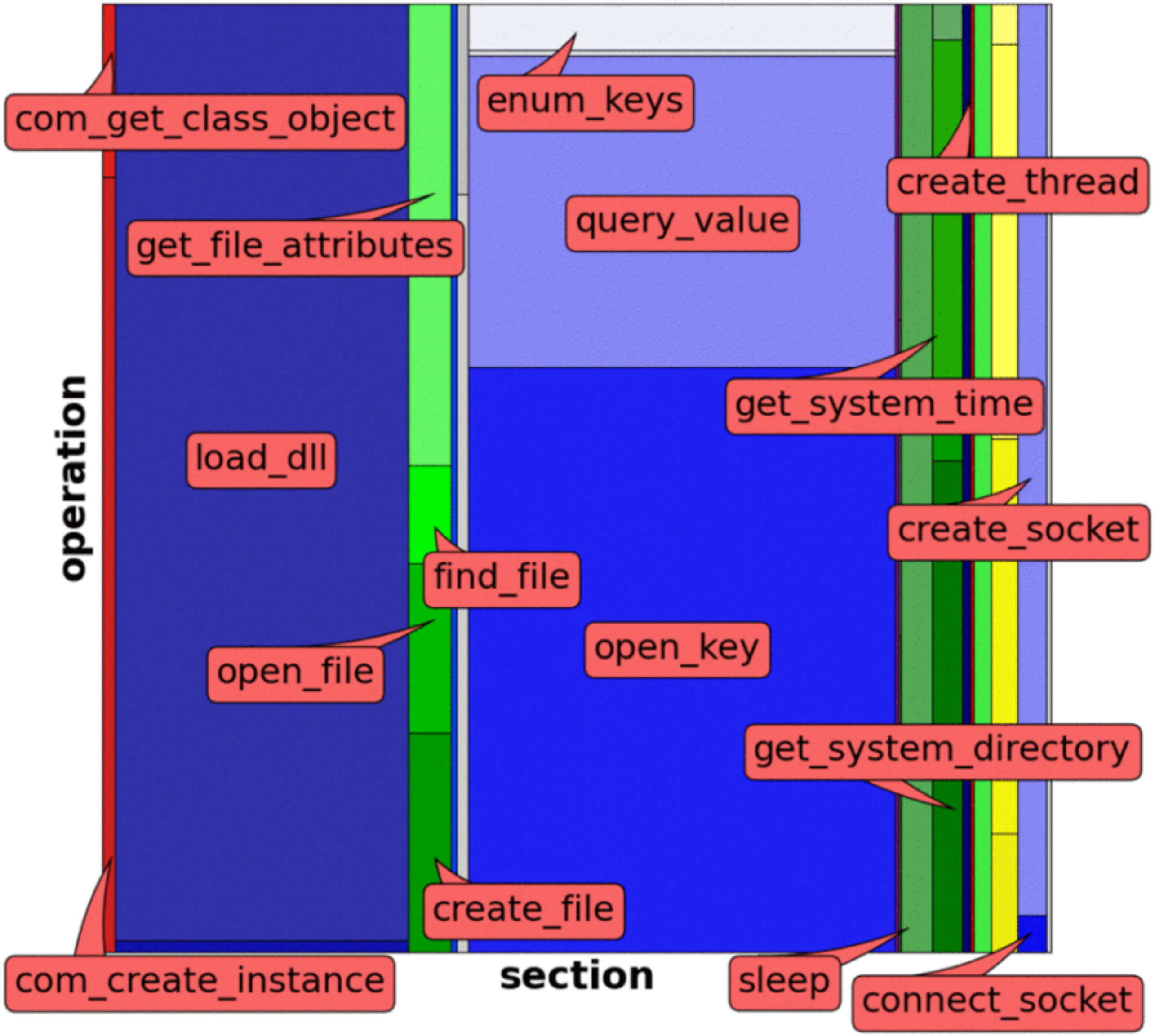
Treemap

- Nested rectangles
  - Width proportional to percentage of section’s API calls
  - Height proportional to API operation frequency
- Sections are plotted in fixed colors and order

Treemap Colors, Left to Right

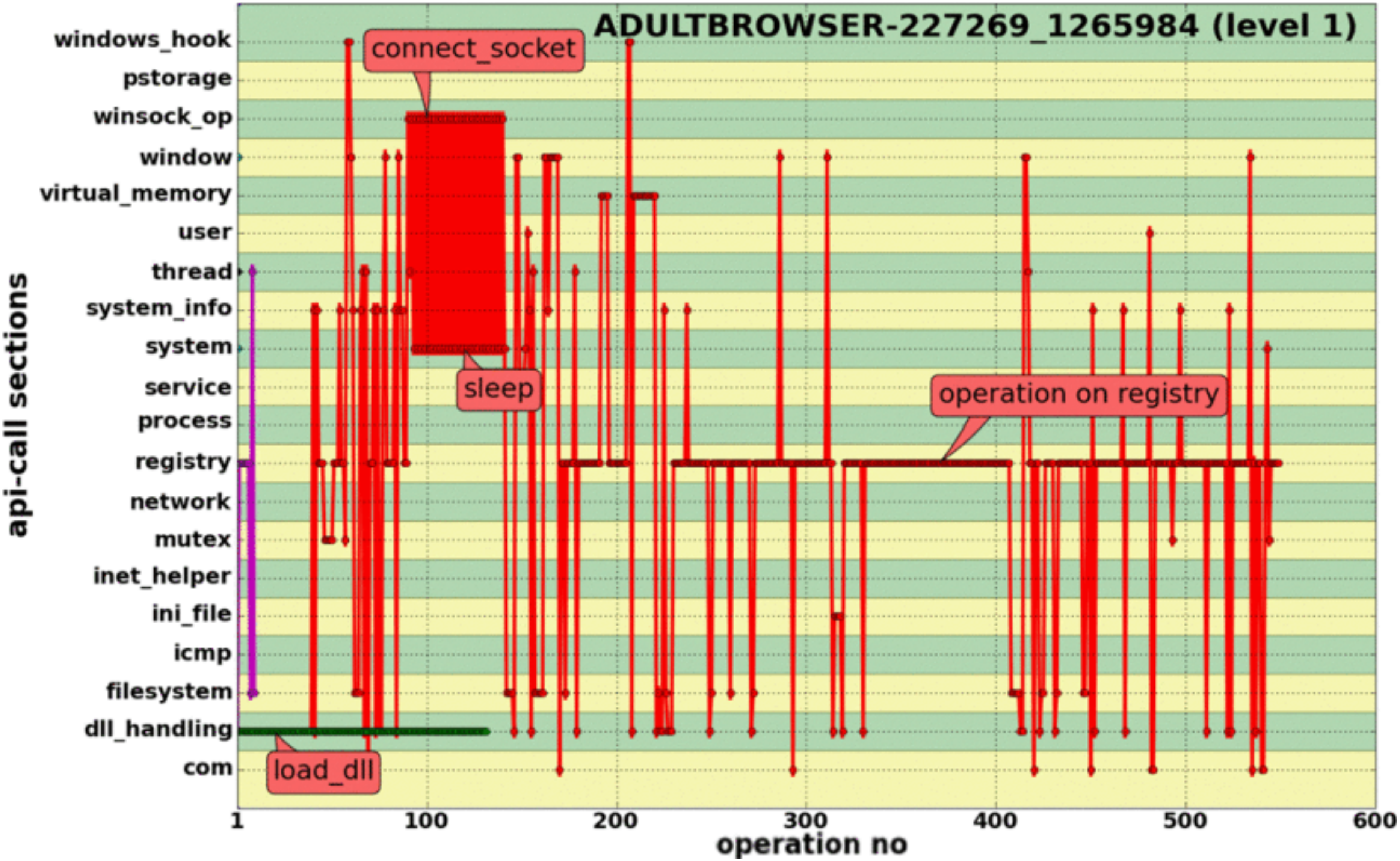
Color	Section
Red	Com
Blue	DLL handling
Green / Light Green	File system
Royal Blue	Ini
Dark Blue / Purple	Registry
Magenta / Grey / Light Green	Process info
Green	System info

ADULTBROWSER-227269_1265984

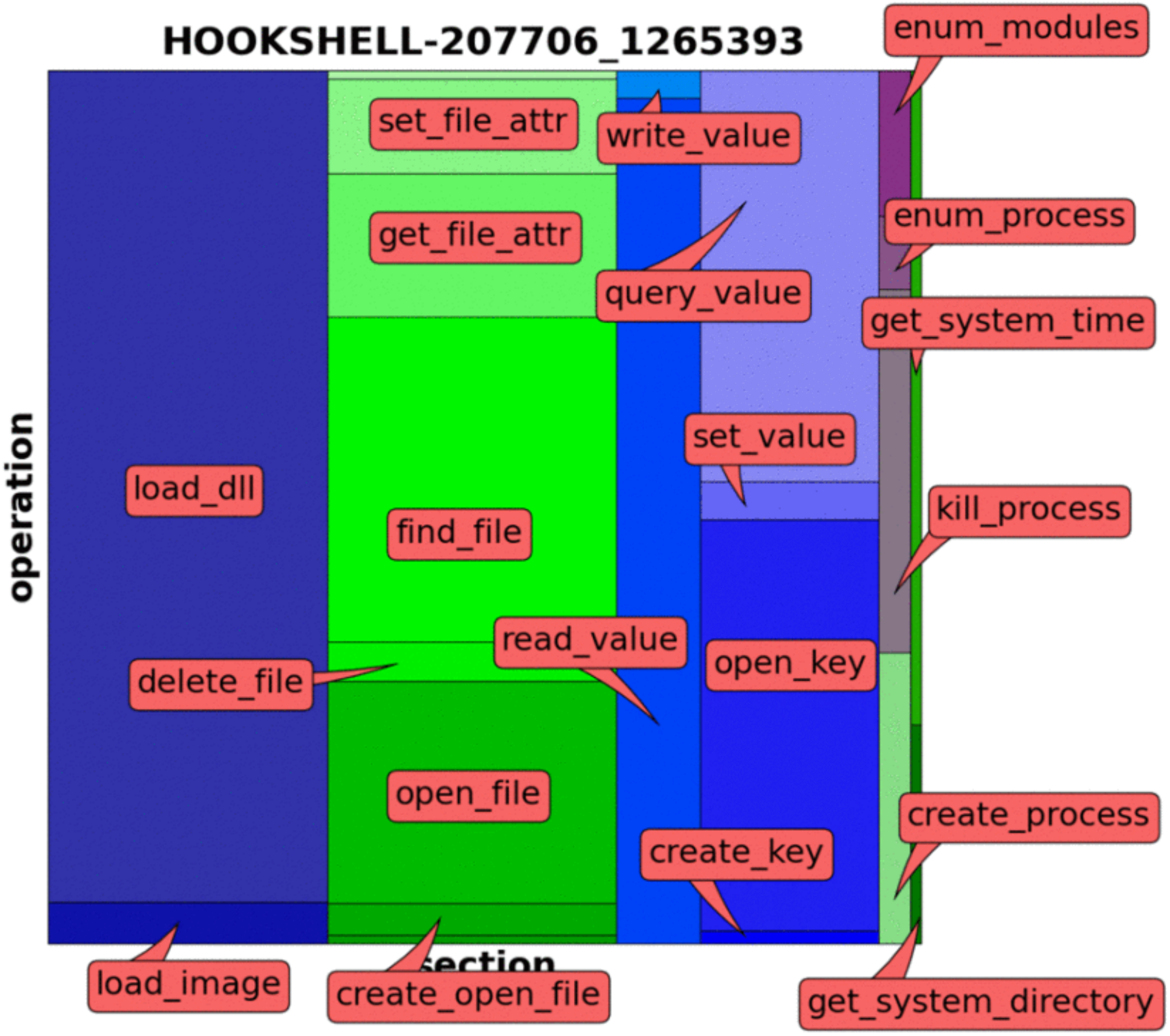


Thread Graph

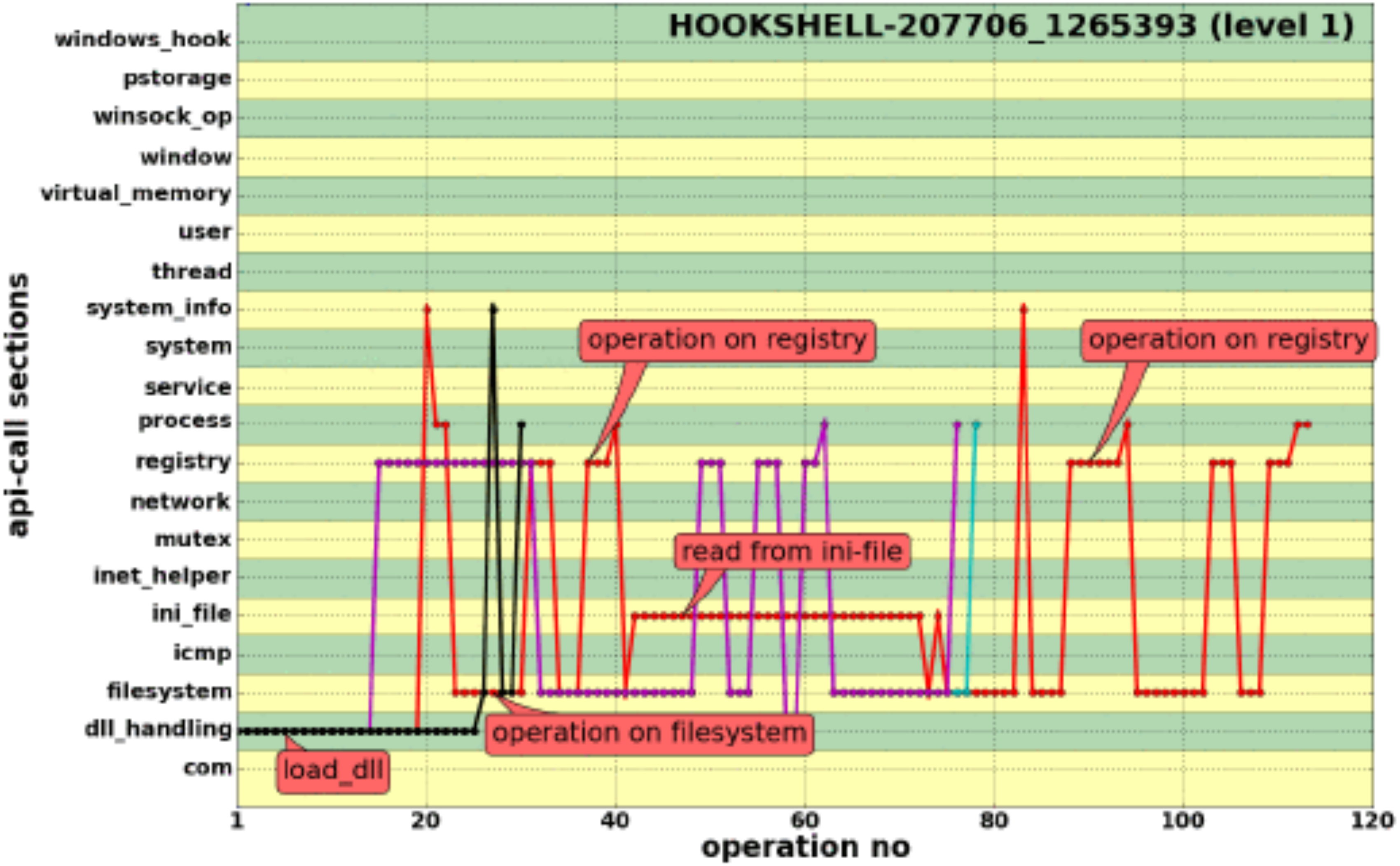
- X axis is time
- Y indicates section or operation
- Different threads are shown in different colors
  - Operations have to exceed a threshold to be displayed


ADULTBROWSER-227269_1265984 (level 1)
api-call sections
windows_hook
pstorage
winsock_op
window
virtual_memory
user
thread
system_info
system
service
process
registry
network
mutex
inet_helper
ini_file
icmp
filesystem
dll_handling
com
connect_socket
sleep
operation on registry
load_dll
operation no
1
100
200
300
400
500
600

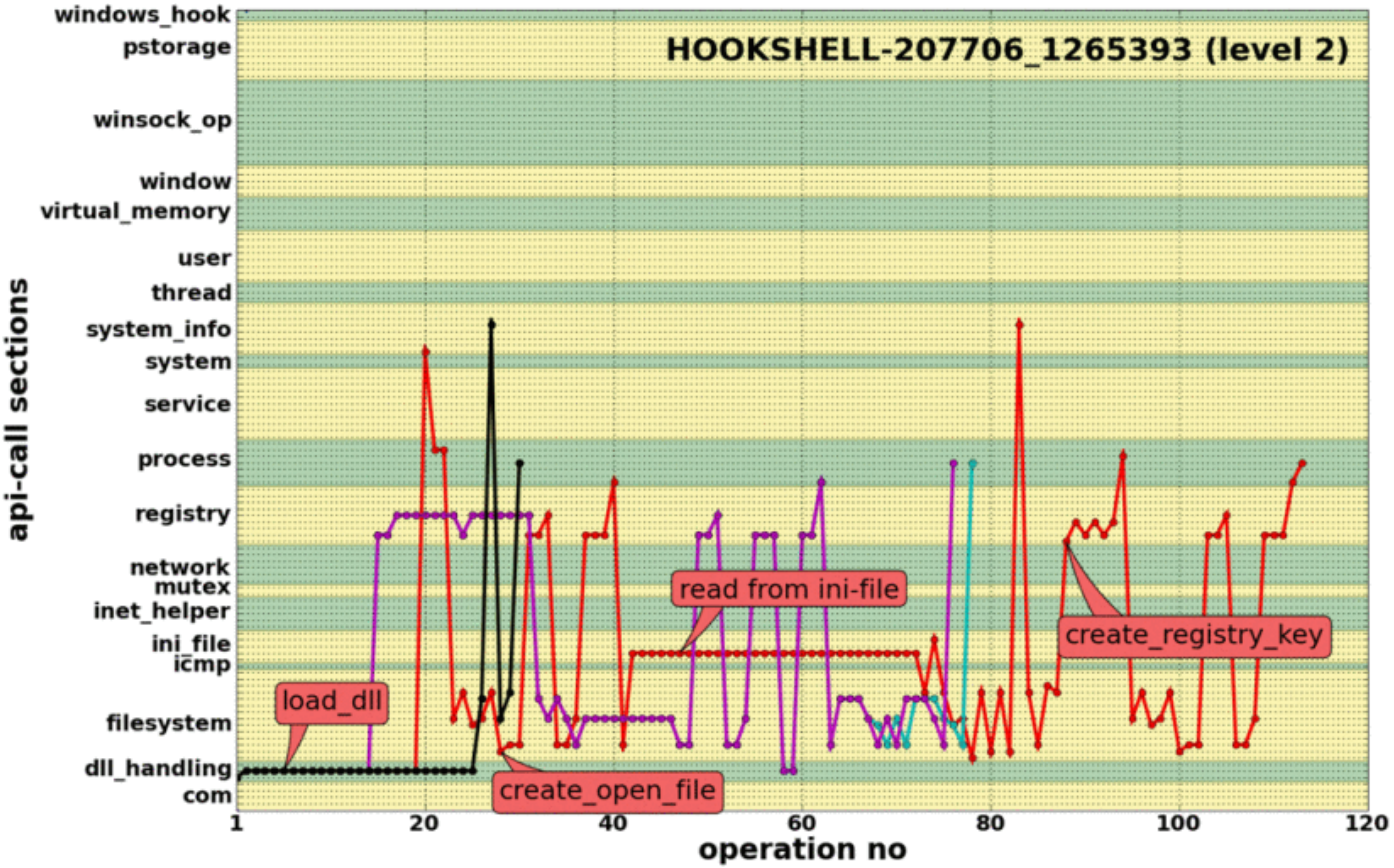
Hookshell Treemap



Hookshell – sections only



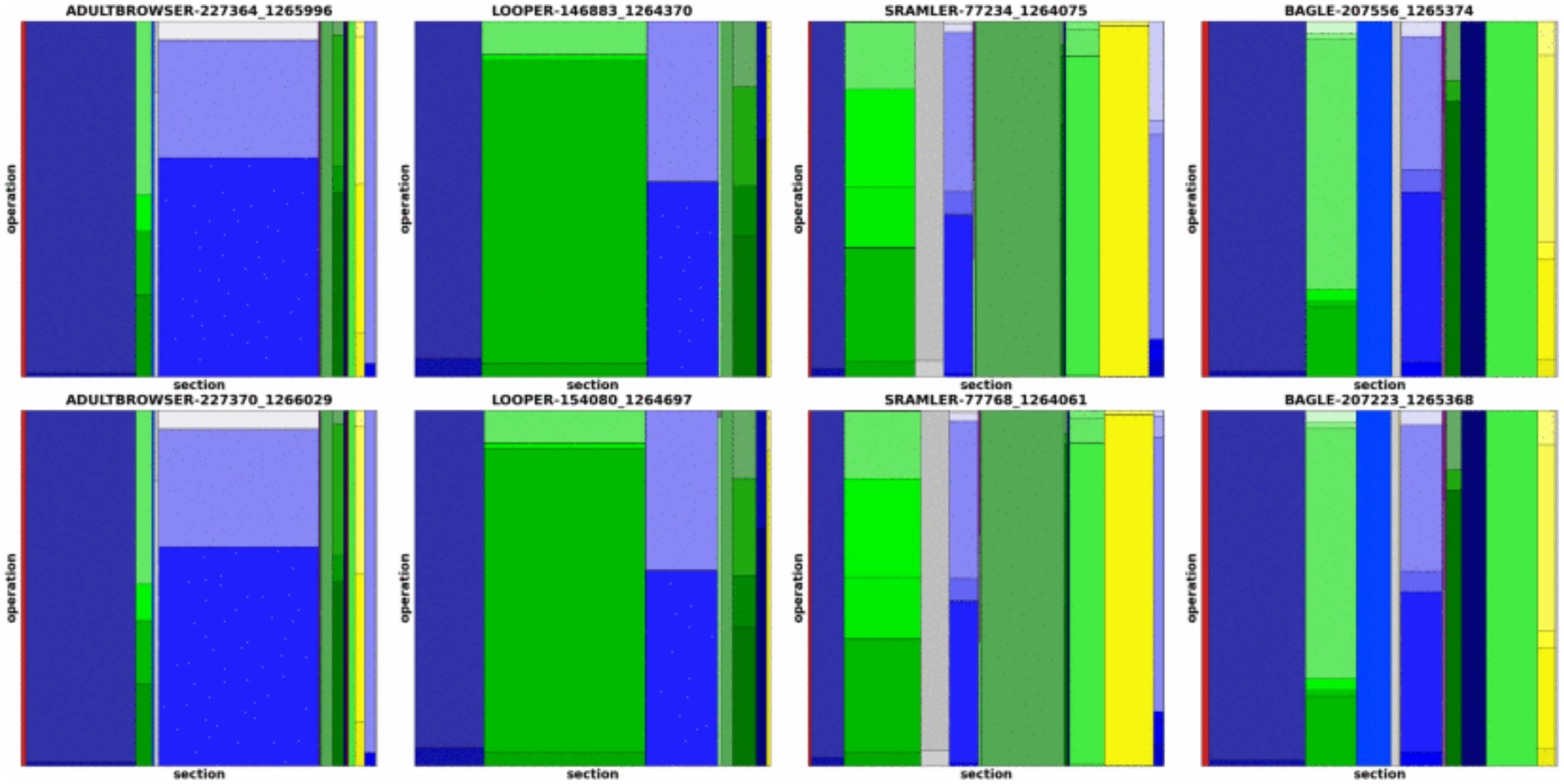
Hookshell – sections + operations



Visual Clustering

- Based on 13 families from 2,000 samples of known malware
- Authors conclude is that visual matching is of limited use

Treemaps by Family



ADULTBROWSER-227364_1265996

LOOPER-146883_1264370

SRAMLER-77234_1264075

BAGLE-207556_1265374

ADULTBROWSER-227370_1266029

LOOPER-154080_1264697

SRAMLER-77768_1264061

BAGLE-207223_1265368

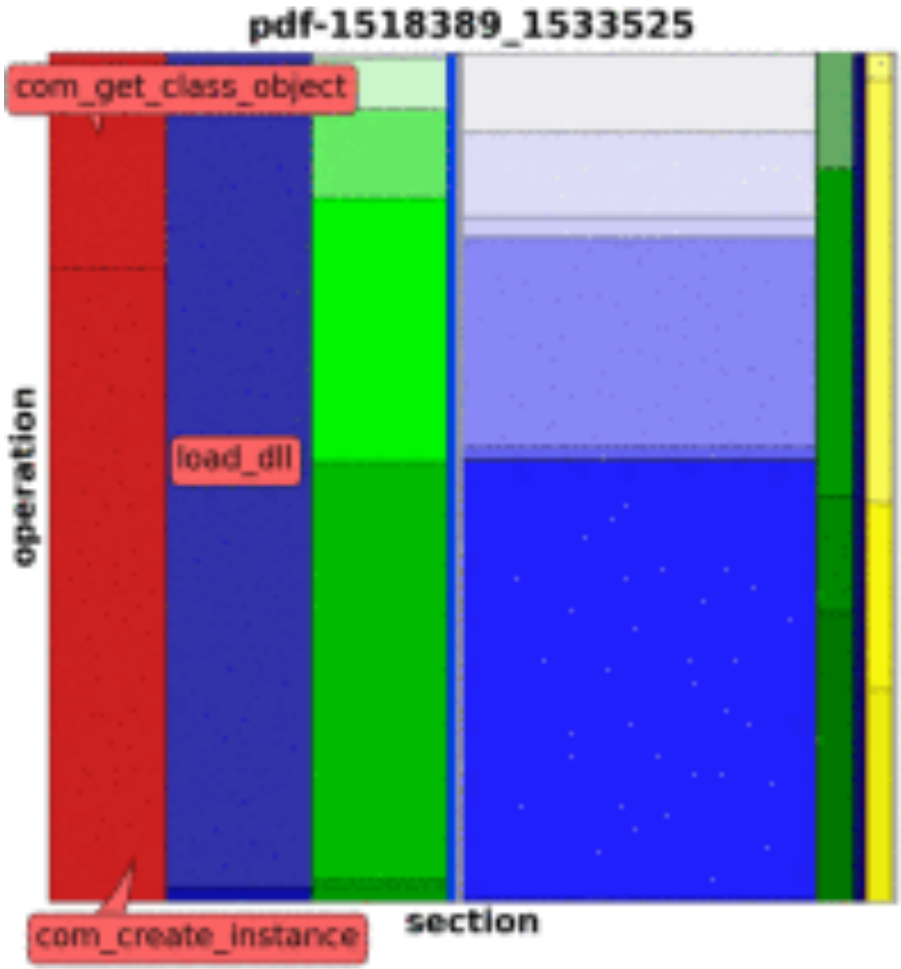
Visualizing Data Files

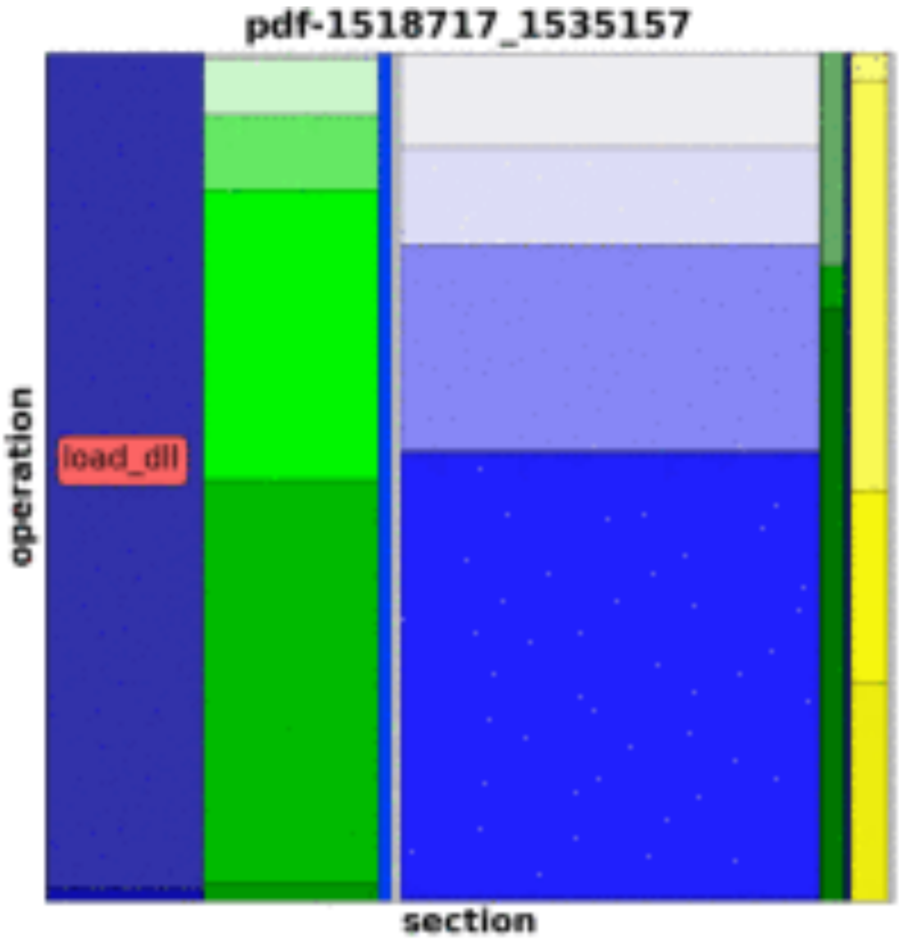
- One can visualize a data file by opening it with its intended application
- Analyzed 17 malicious and 200 benign PDF files

Results

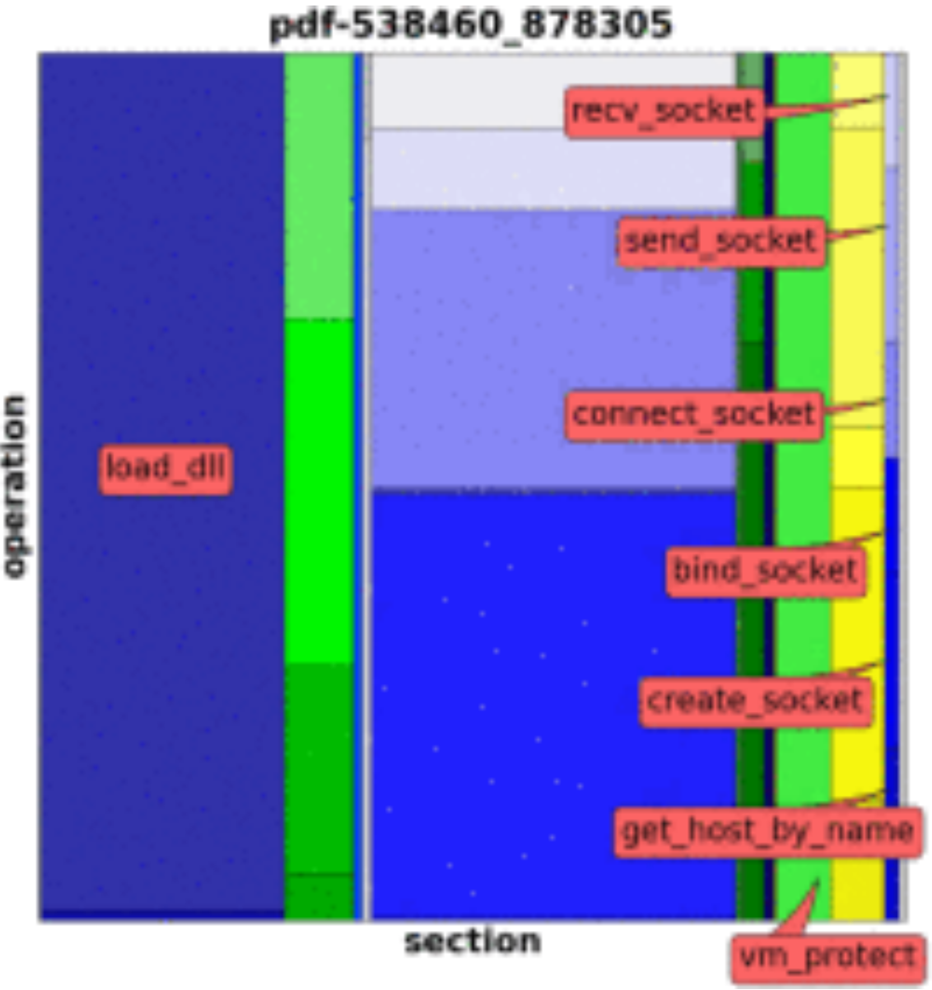
- Adobe Acrobat Reader does not show the same behavior for all the benign PDF files
- Nor does it show the same behavior for all malicious PDF files

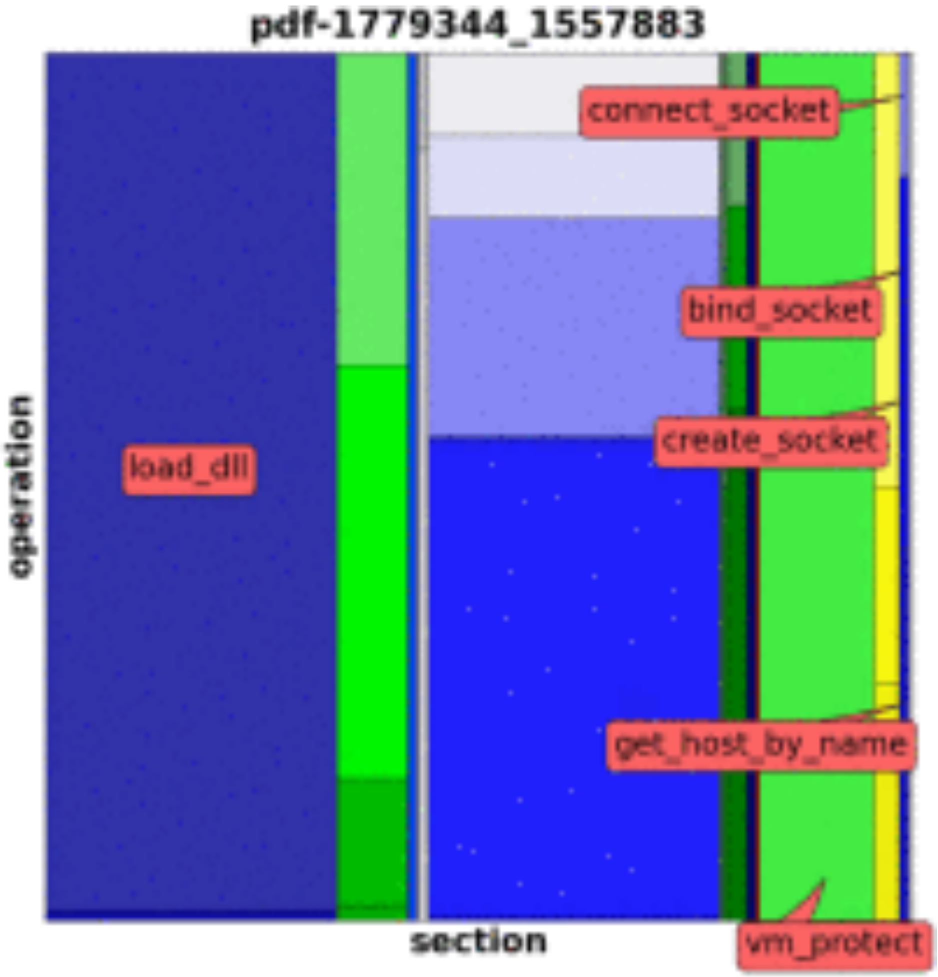
Both Benign PDF



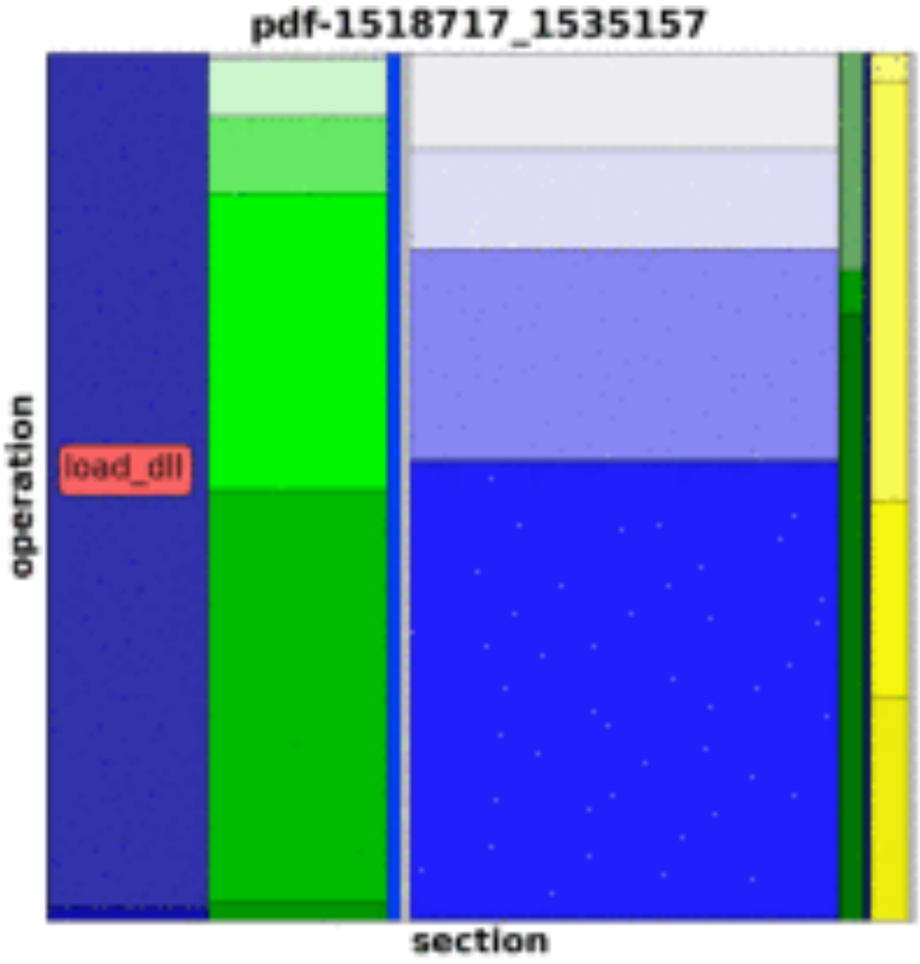


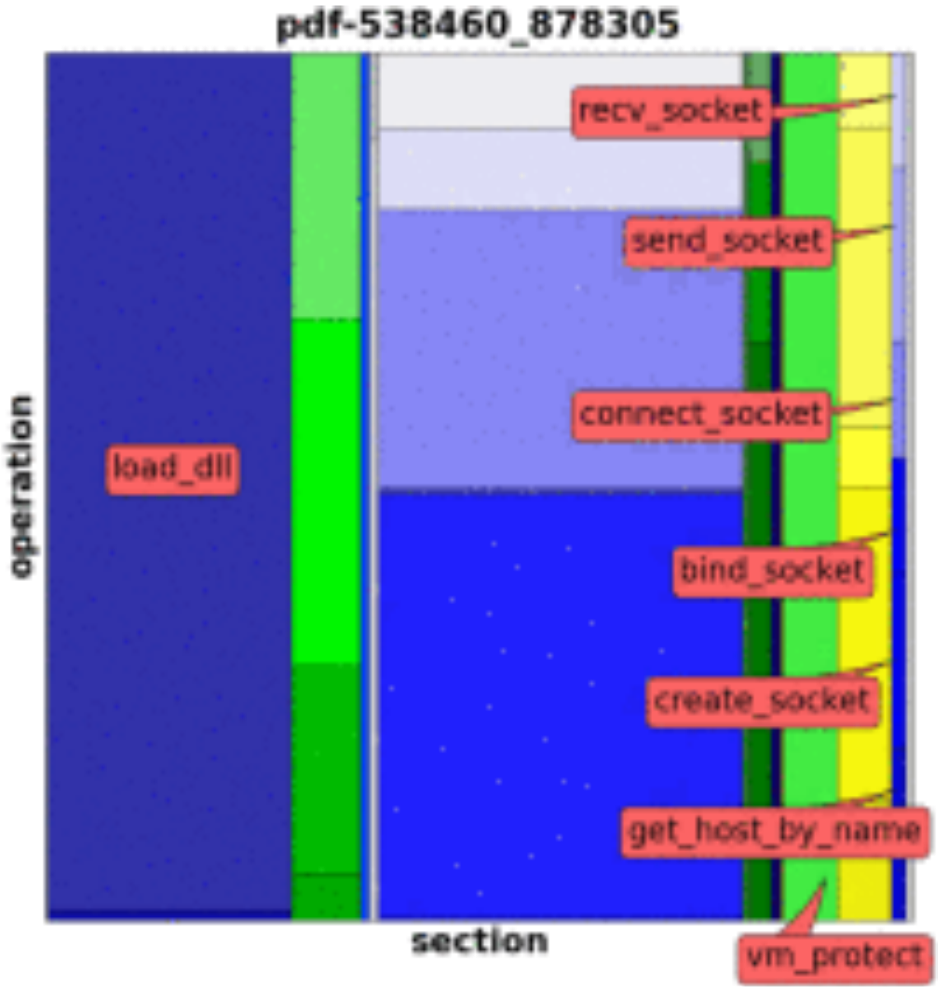
Both Malicious PDF



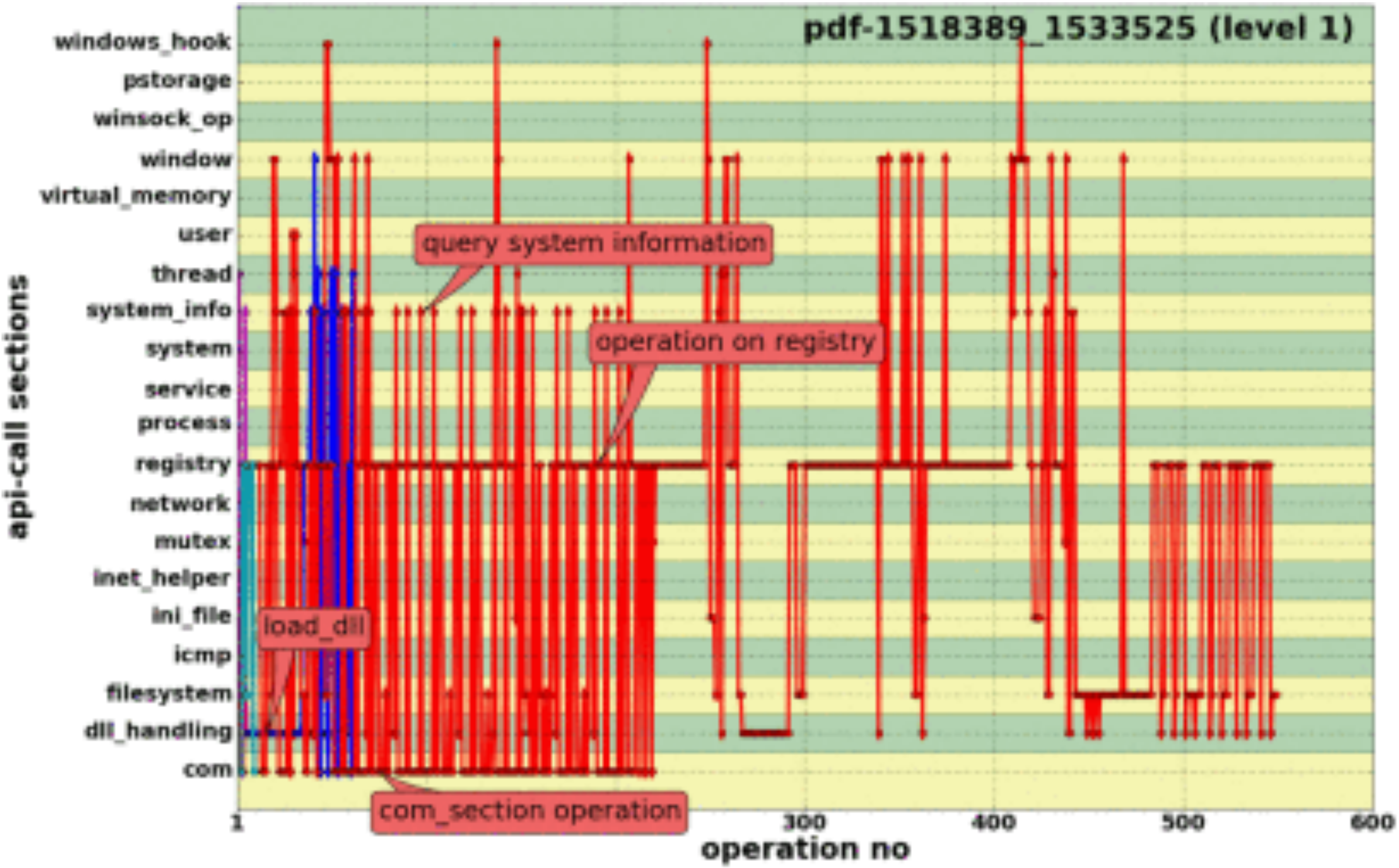


Benign & Malicious PDF





Benign PDF



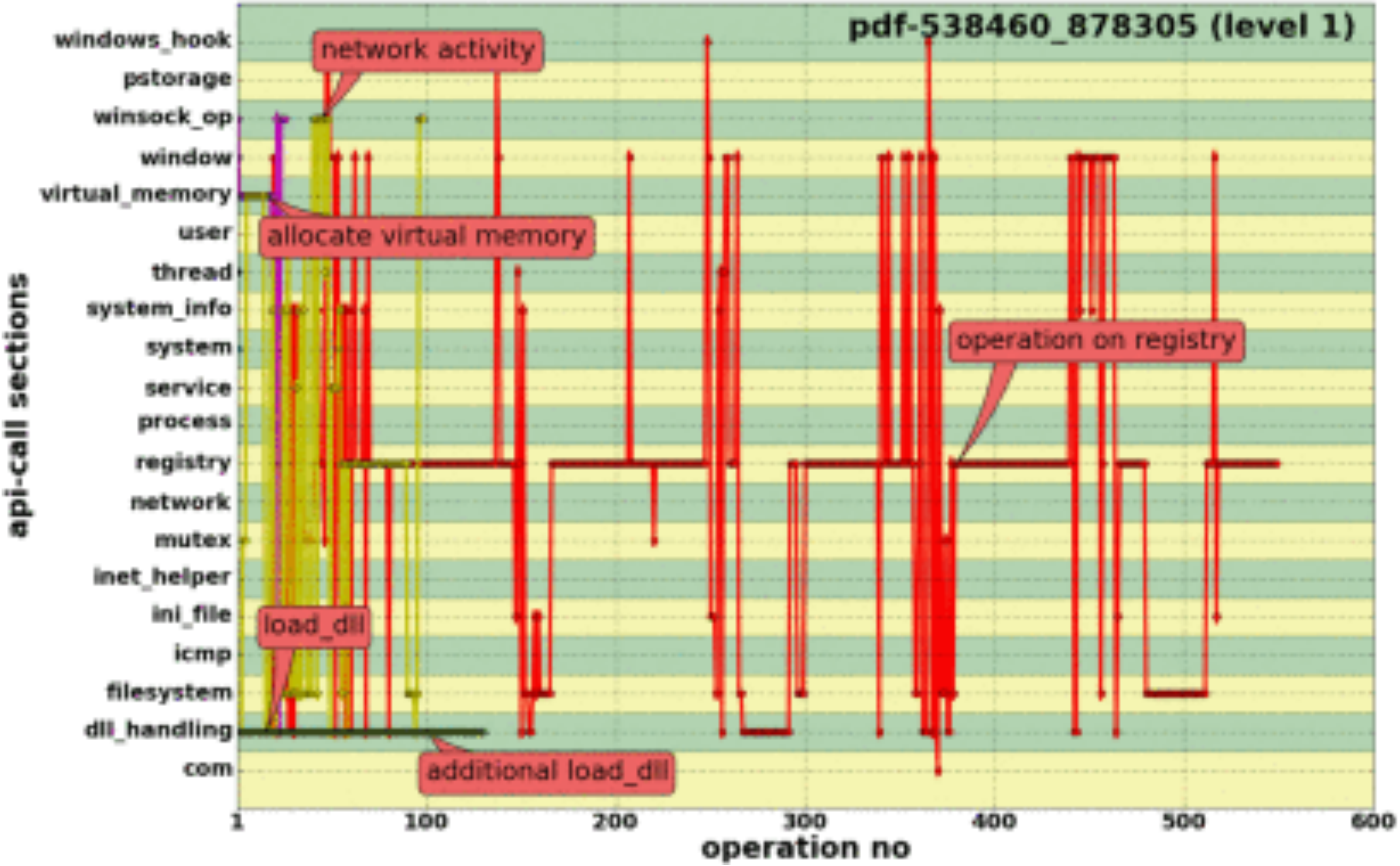
query system information

operation on registry

load_dll

com_section operation

Malicious PDF



network activity

allocate virtual memory

operation on registry

load_dll

additional load_dll

Conclusions

- Treemaps and thread graphs are useful visualizations of malware
- Visual clustering is interesting, but the authors were unclear about its utility