Semantic Binary Exploration

Speeding up malware analysis

Laura Guevara and Daniel Plohmann

laura.guevara@fkie.fraunhofer.de
daniel.plohmann@fkie.fraunhofer.de

Dec. 3rd 2014, Nancy

Fraunhofer FKIE

Outline for this Talk

- Motivation
- Preface: Behaviour Analysis
- Semantics Exploration
  - Malware Semantics
  - Methodology
  - Algorithm
- Demo
  - Introduction to IDAscope
  - Semantic Explorer vs. Citadel
- Conclusion

Motivation

Why build a scanner for semantic exploration

- Experiences of daily work:
  - „What are the capabilities of this unknown executable?"

- Observation: Different malware samples share many common aspects of malicious functionality
  - Evolution of version within one family: minor modification or changed appearance through compiler fragmentation
  - Authors seem to have copy-cat mentality regarding snippets available on the Internet

Static Analysis

- Decoupling analysis from the malware's execution time
  - Access all of the code (also „dormant" parts)
  - Allows exploration and documentation at the same time

- Automated tool
  - Explore the control flow graph of executable Windows memory images
  - Support the analyst during static analysis
  - Guidence to specific regions of interest

Our Approach

- We examine sequences of calls to API functions in malware instances and try to infer the user-level functionality connected to them.

Malware Behaviour Patterns

Malware Features

- Equivalent features reappear from one malware variant to another
  - Shaped differently in code
  - But with a predictable occurrence of used API functions
- Abstracting the interaction of malware with the Operating System
  - Syntax vs. Semantics
- Platform dependent
  - Windows Subsystem DLLs

Windows API

Windows Application Programming Interface

Windows API

- Formerly called Win32 API
- Specifies a collection of services needed during runtime
- Usually loaded before or during the actual execution
- A common way to analyze the behaviour of programs is by inspecting its calls to API functions

Abstracting Behaviour

Malware Semantics

Semantics

„Behaviour Profile“

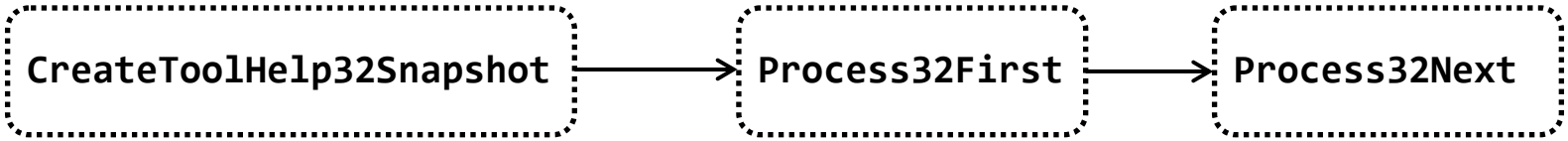
- Assign meaning to the set of common malware operations
  - Copying or deleting files for hidden persistence
  - Injecting into processes for more control or concealment
  - Communicating over the network, etc.
- These are usually implemented using calls to a specific collection of Windows API subroutines.

Malware Semantics

Example

Process Injection

Step 1. Iteration over Processes



Step 2. Process injection

Overview

Semantic Binary Exploration

Import Table Directory

- Our tool requires availability of API information (e.g. restored import tables) since resolved call destinations are cataloged and examined

Unpacked Binaries

- Applicable to files in Windows Portable Executable (PE) format and shellcode

Methodology

- Collection of Malware Behaviour
- Definition of malware semantics
- Exploration:
  - Extraction of flow components
  - Call Graph construction
  - Matching of specifications
  - Semantic Traces:
  - Cross-evaluation with HCA

Semantic Explorer

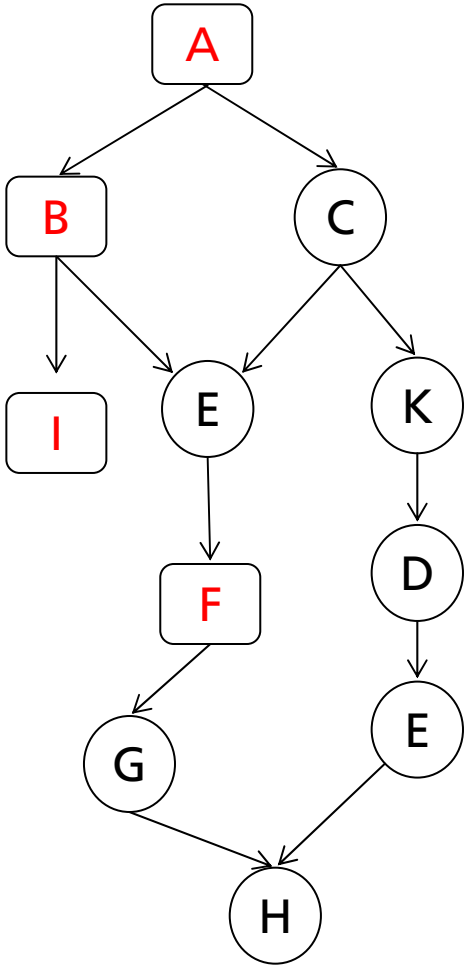
Argument Parsing

- Backtrace reference of registers
- String Parsing
  - Based on Alexander Hanel's work

Difficulty:

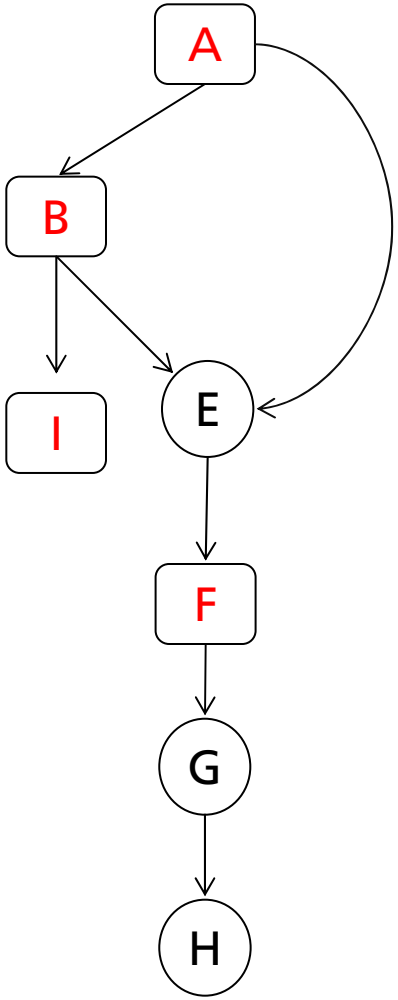
- Address every possible scenario w.r.t how the data is moved
  - Cross-reference
  - Function return values

Data Reduction Strategies



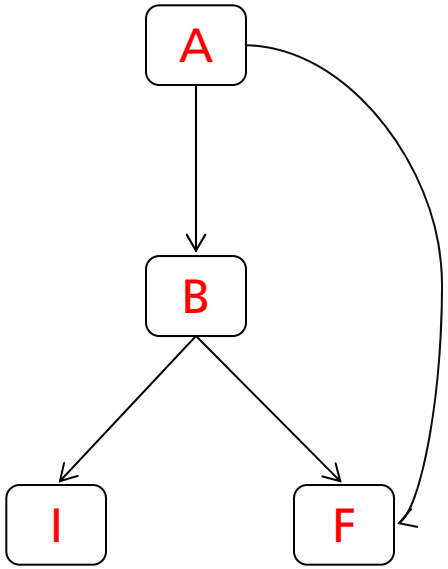
{ A, B, I, F }

Data Reduction Strategies



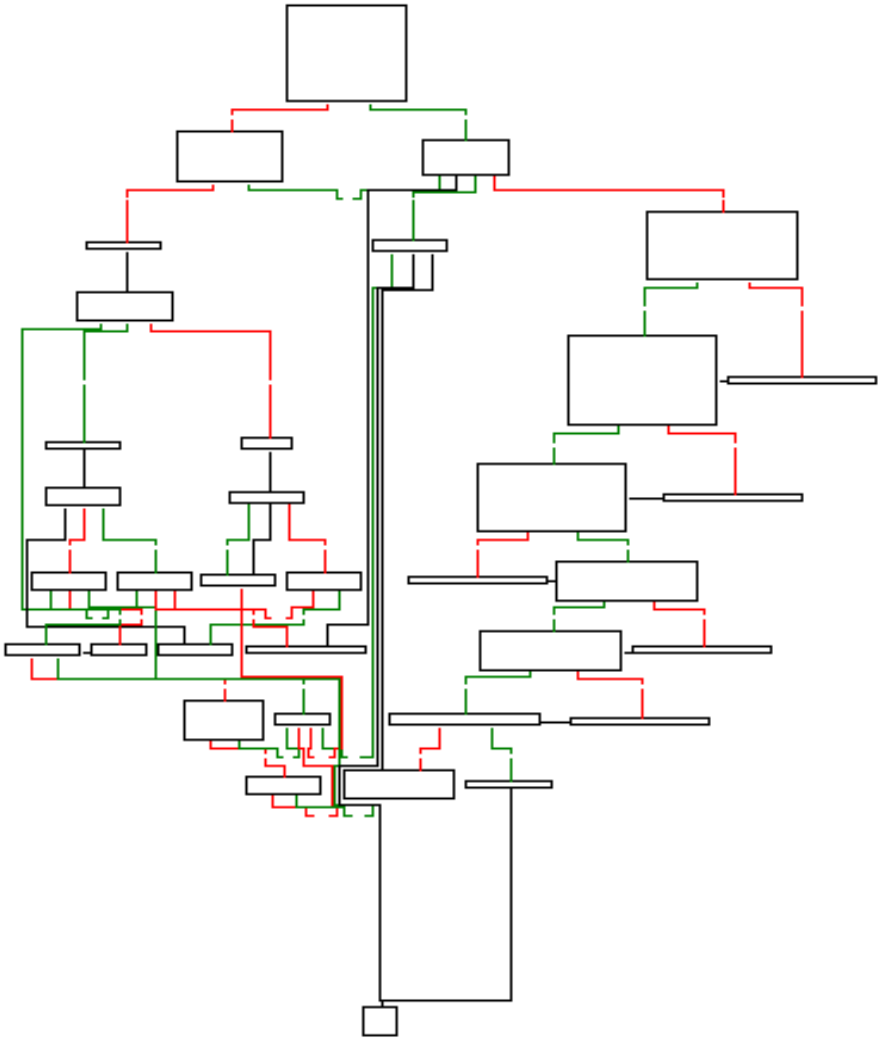
{ A, B, I, F }

Data Reduction Strategies

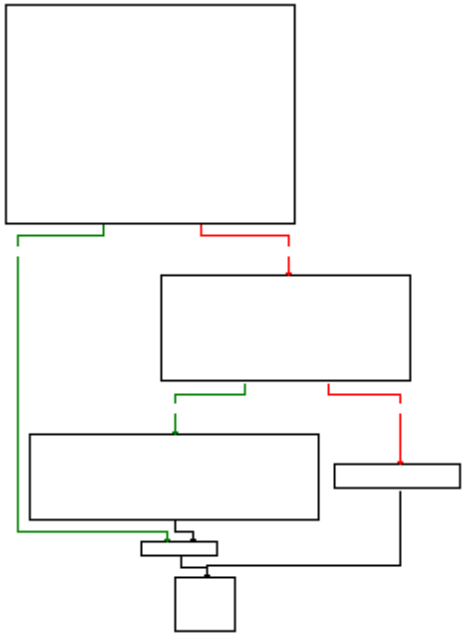


{ A, B, I, F }

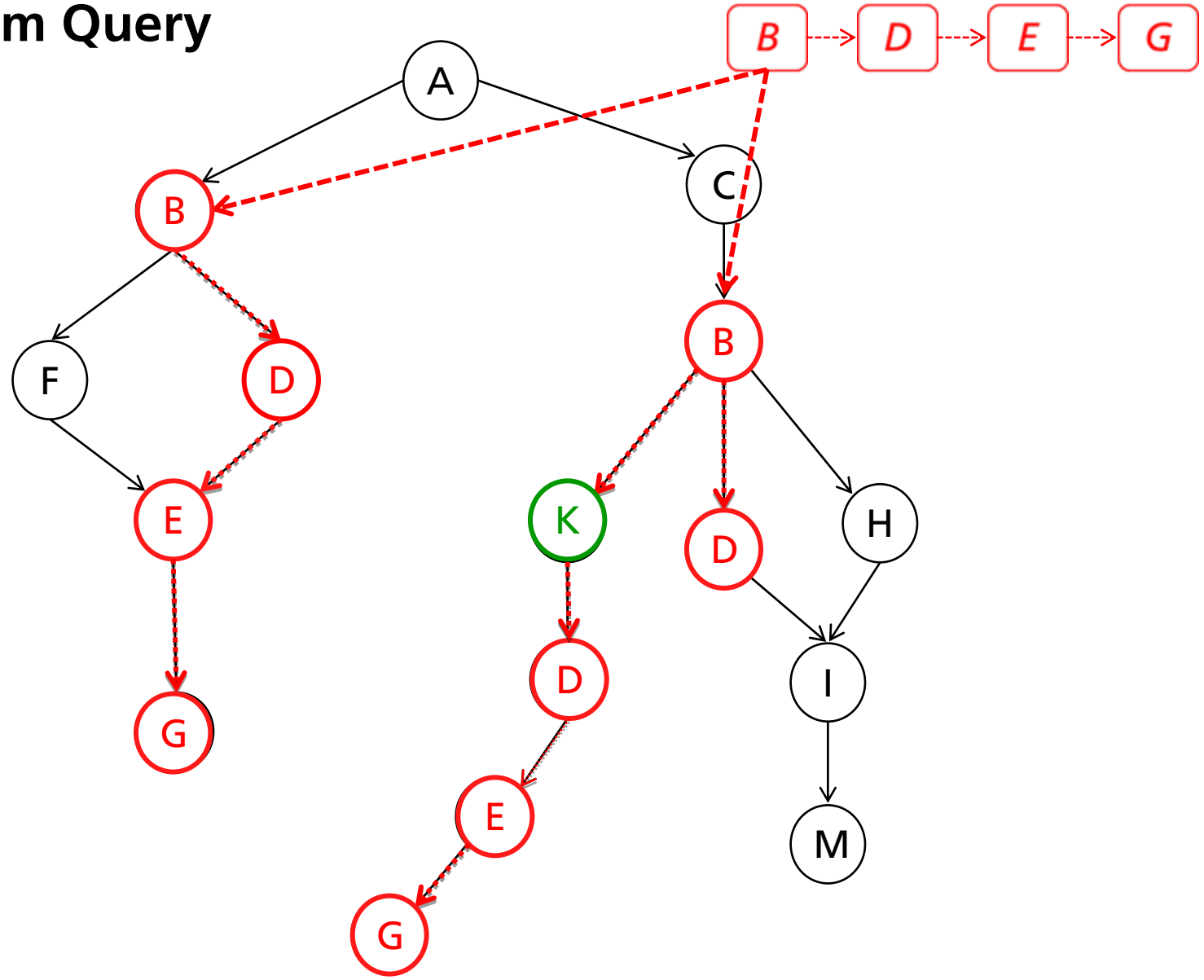
Data Reduction Strategies



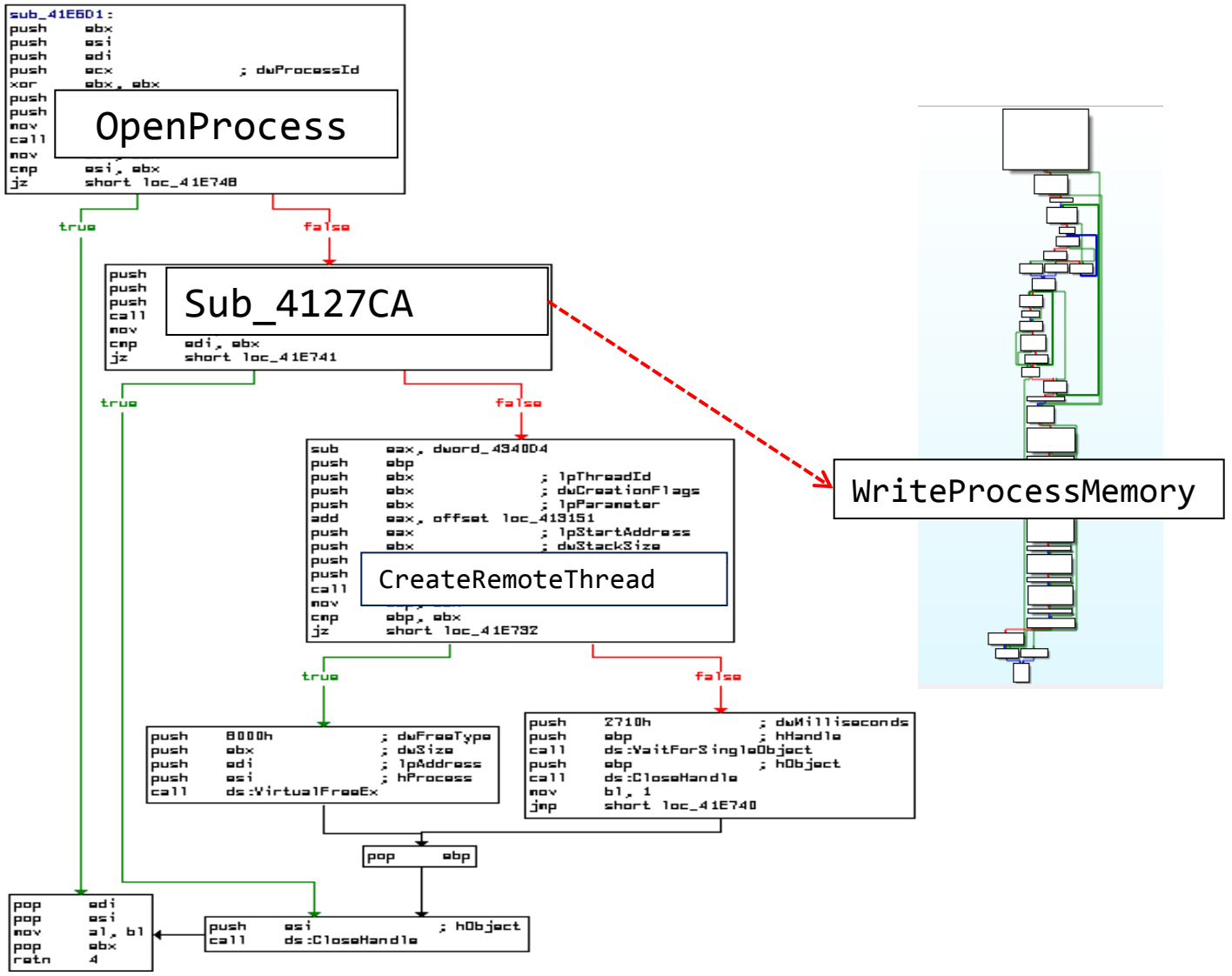
Data Reduction Strategies



N-Gram Query



Nested Functionality and Control Flow Integrity



The following slides reflect the content of the

Demo

IDAscope

An IDA Pro extension to aid malware reverse engineering

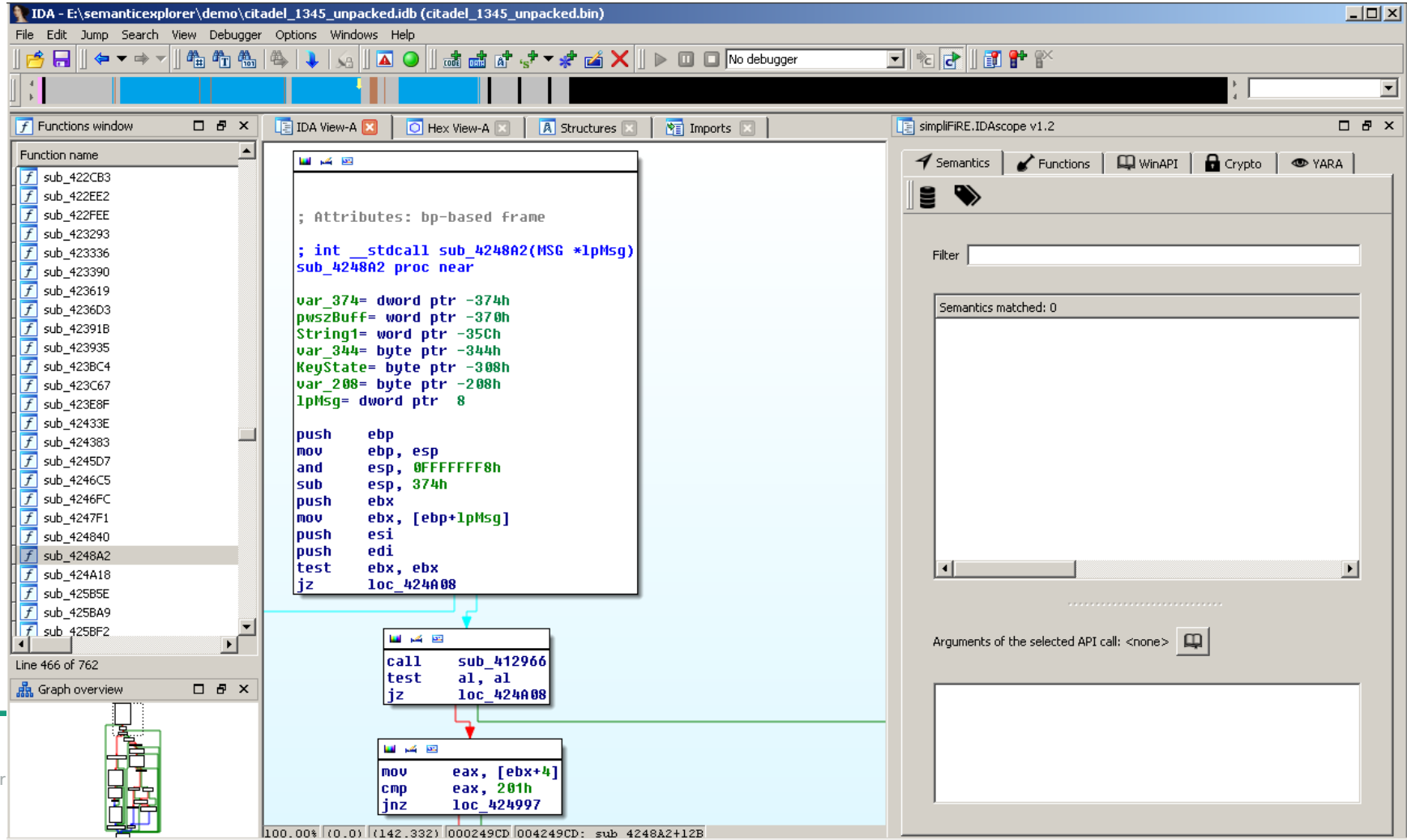
- Motivated by the current typical workflow of working with IDA Pro.
  - Repeat: „Identify relevant parts of the binary; tear apart; document findings.“
- Common tasks:
  - Work corner pieces: strings, API calls, signature hits, …
  - Reoccurring need for looking up things in MSDN (switch windows…)
  - C&C communication schemes are of high interest!
  - Find and understand cryptographic routines used.
- Idea:
  - Provide automation/integration of „helpers“ that assist with regularly performed tasks.

Fraunhofer FKIE

IDAscope

Seamless integration

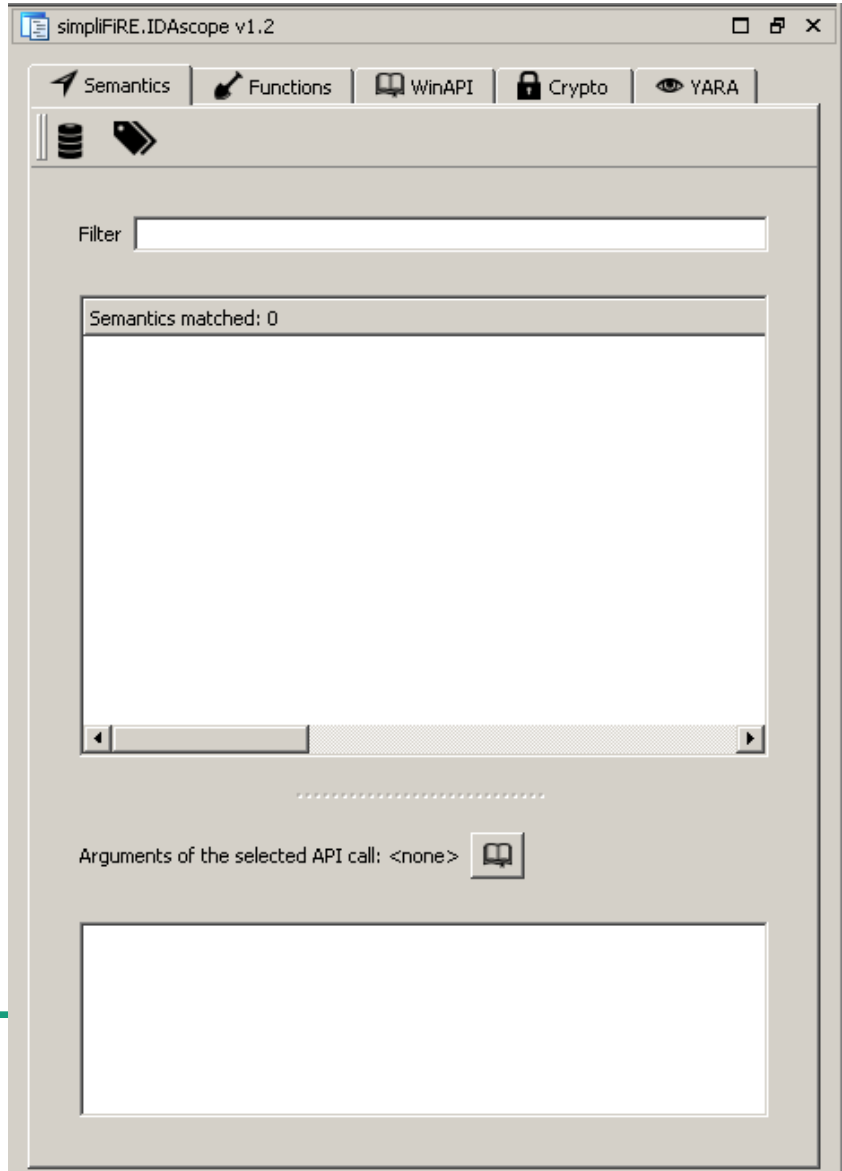
- IDAScope is directly integrated as widget in the IDA interface



IDAscope

… has grown since ist original release (+YARA and Semantic Explorer)

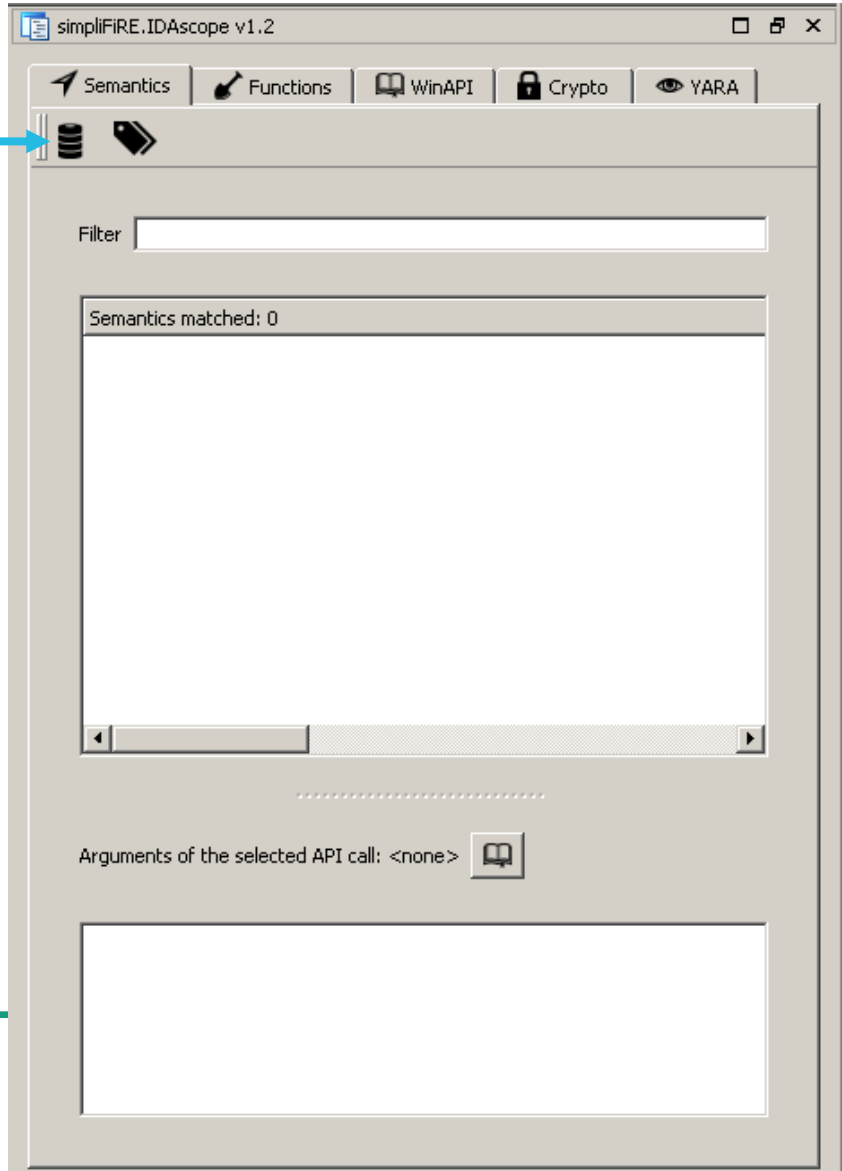
- Tabs for different functions:
  - Semantic Explorer
    - as presented in this talk
  - Function Inspection
    - predecessor for the work presented in this talk
  - MSDN Browsing (WinAPI)
    - seamless lookup of function signatures, enums, …
  - Cryptography Scanner
    - heuristically over instruction type frequency (arithmetic & logic vs. other)
    - signatures for common algorithms
  - YARA Scanner
    - shows incomplete matches, which is useful when writing signatures



IDAscope

Semantic Explorer usage examples (scanning)

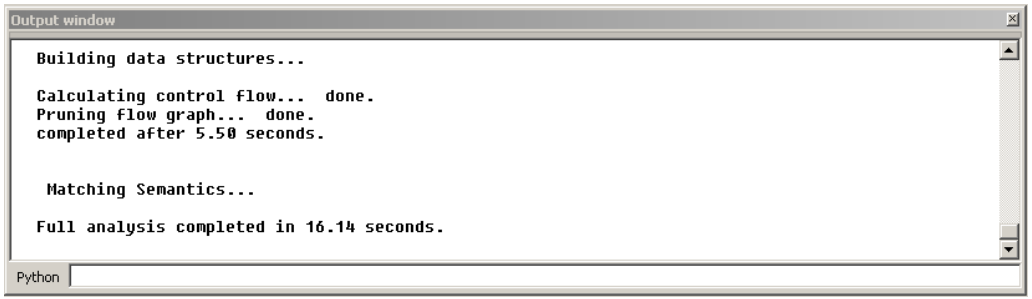
- Clicking the DB symbol will initiate semantic matching

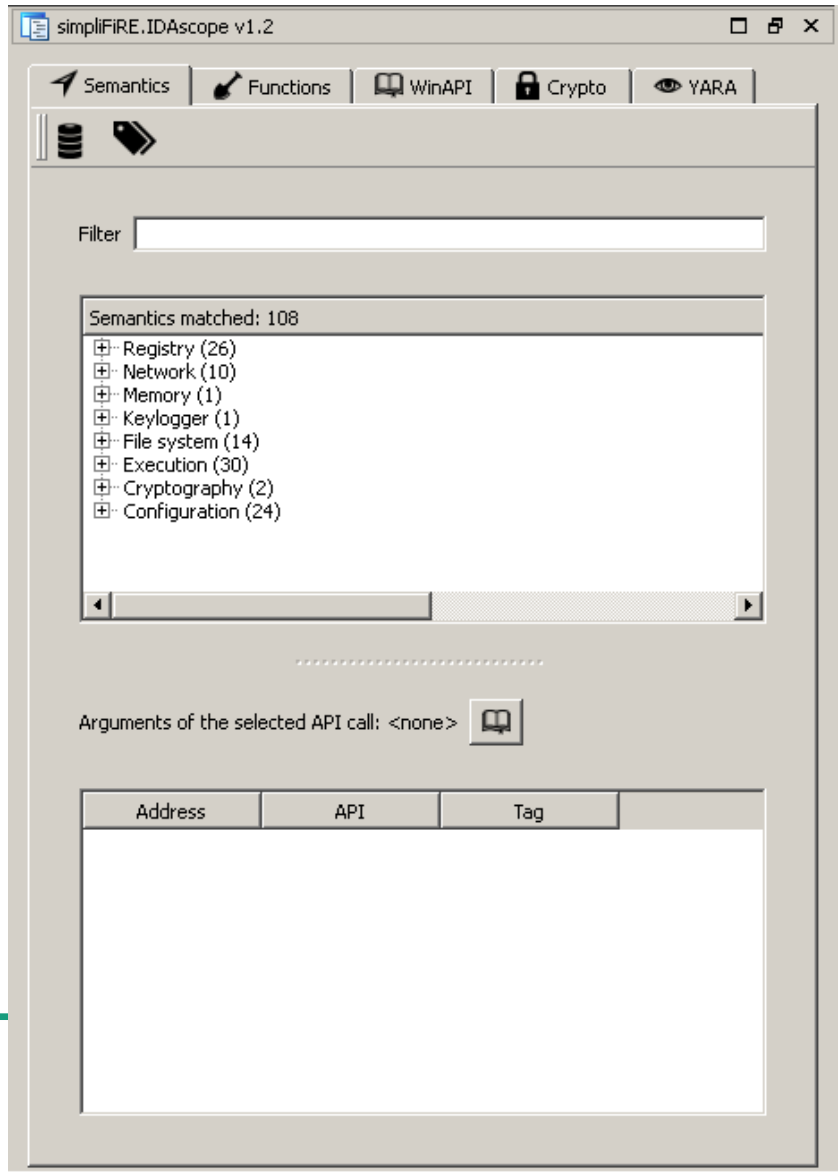


IDAscope

Semantic Explorer usage examples (result display)

- In this example, we have 108 occurrences of matched semantics
- Organized in thematic groups
- These can be freely defined in the config file
- Scanning takes around 20 seconds (sample with 750+ functions)

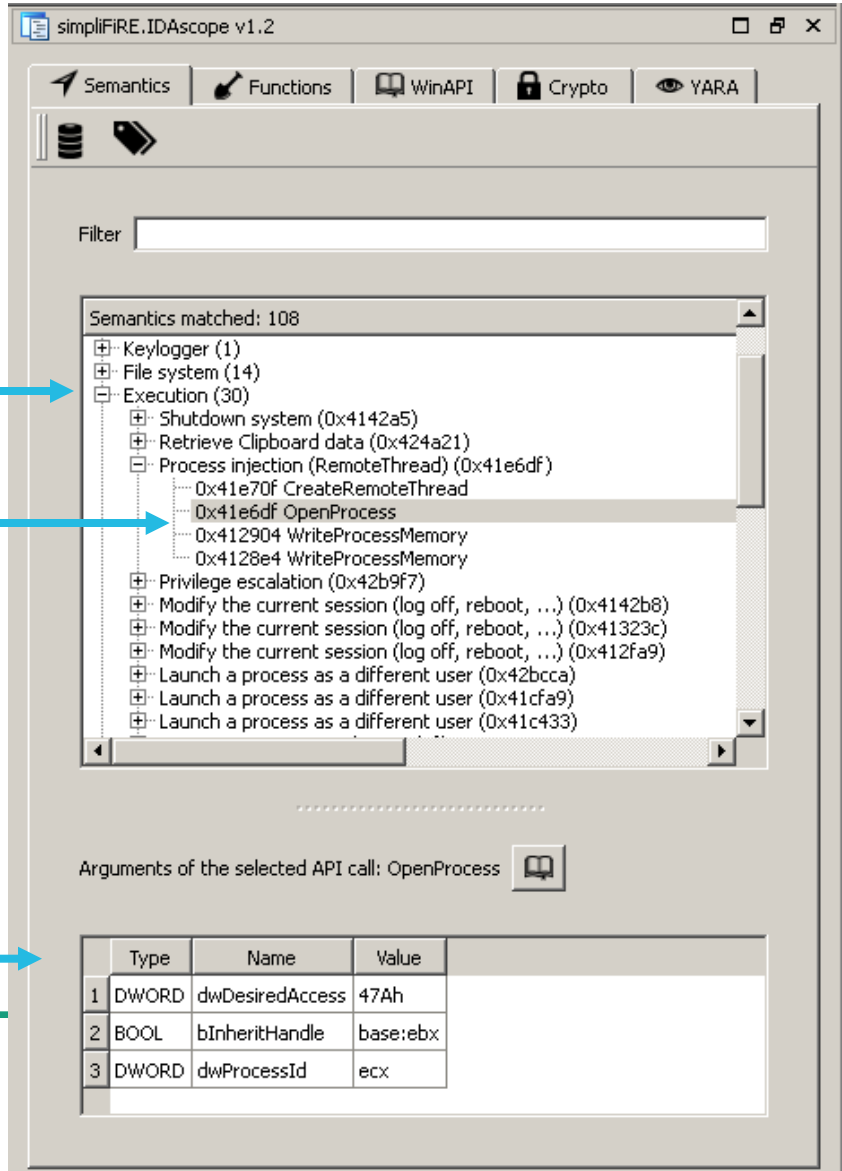




IDAscope

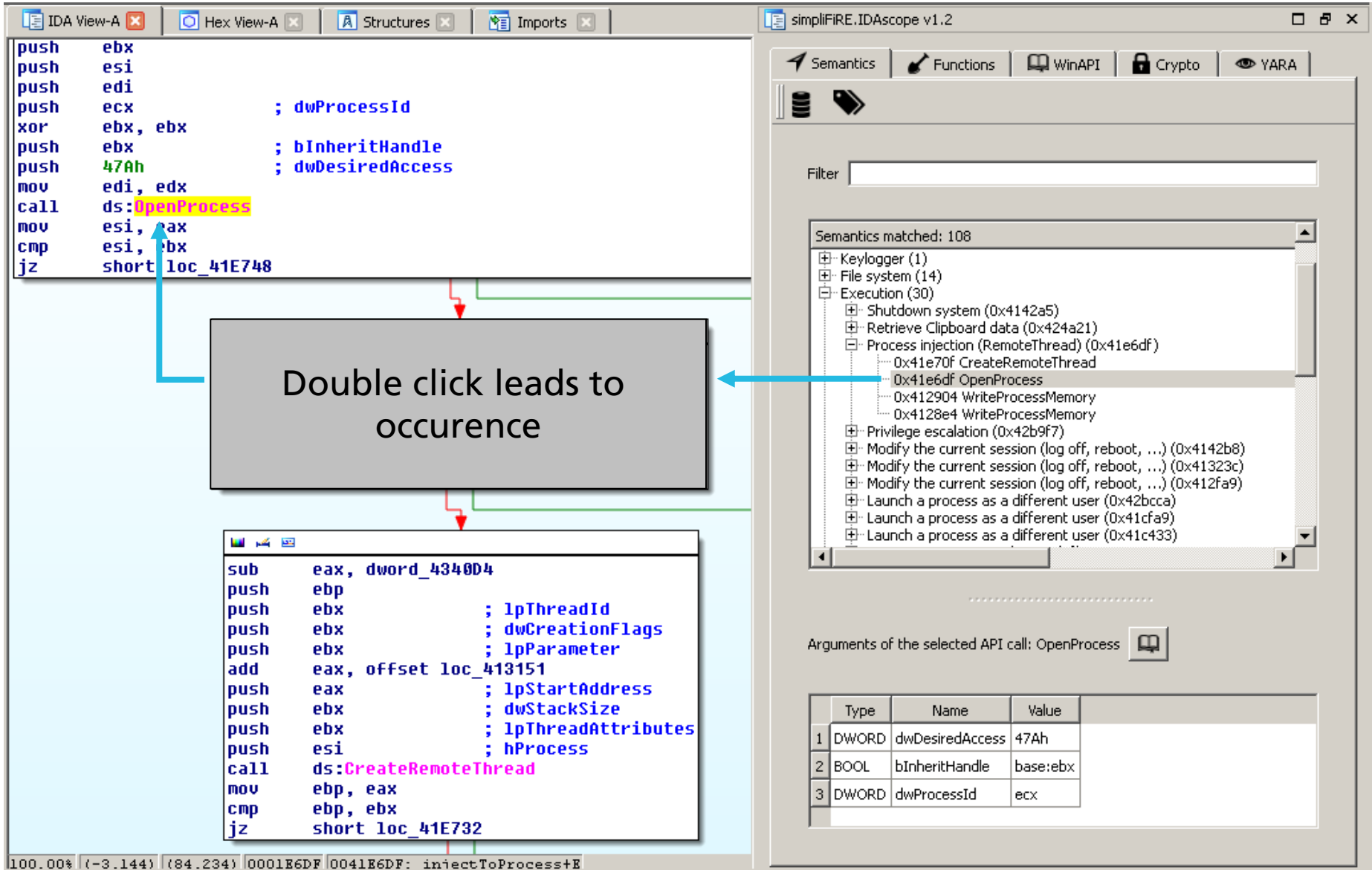
Semantic Explorer usage examples (expanded results)

- Opening one of the groups shows all the semantics contained in the group
- Opening a semantic match shows the suspicious API call sequence
- Selecting an API call shows its arguments



IDAscope

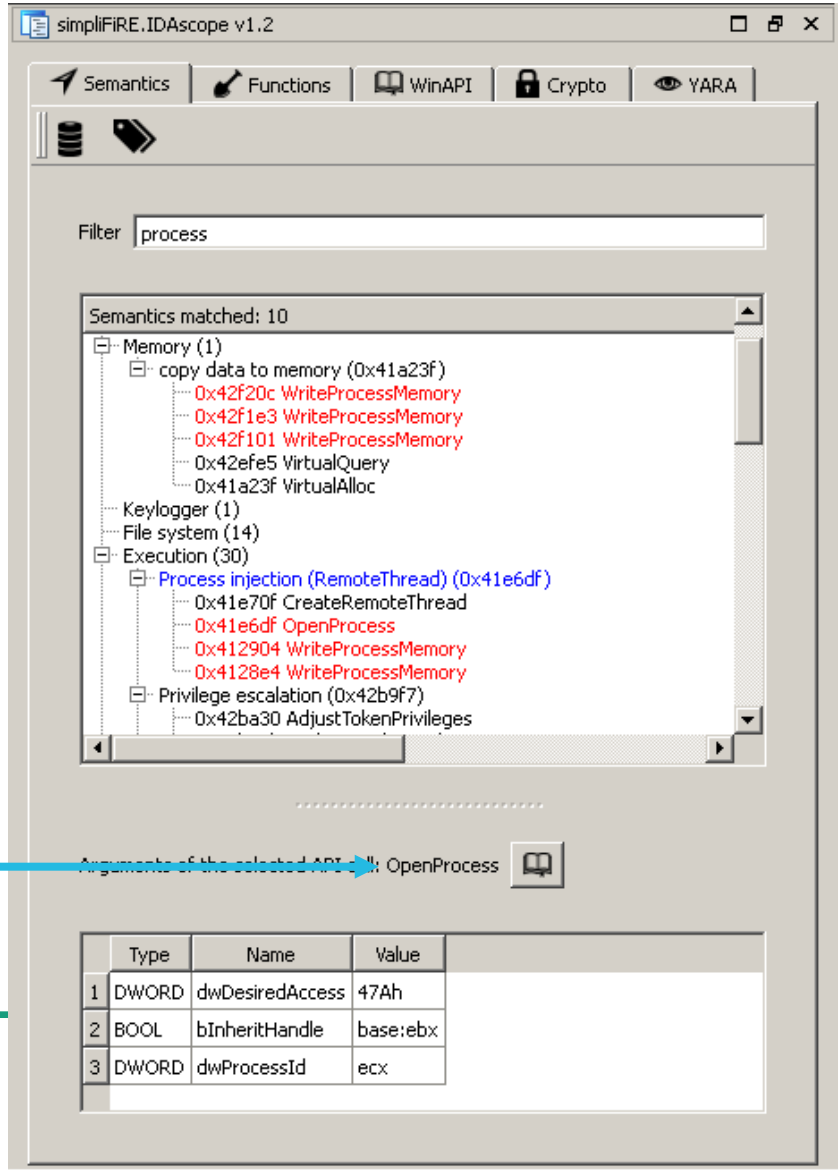
Semantic Explorer usage examples (interactive link to IDA navigation)



IDAscope

Semantic Explorer integration with other tabs (jump to API info)

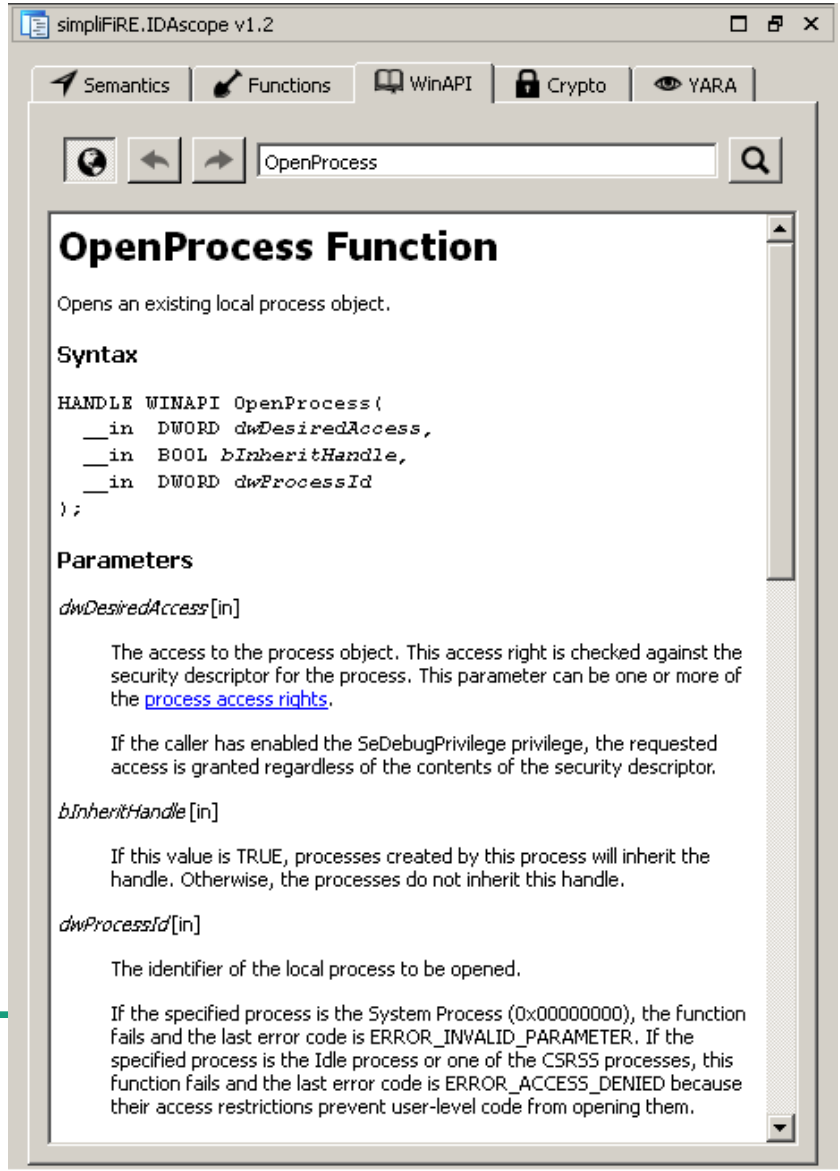
- Clicking the “book” button opens the respective API information in WinAPI view (MSDN entry)



IDAscope

Semantic Explorer integration with other tabs

- Clicking the “book” button opens the respective API information in WinAPI view (MSDN entry)



IDAscope & Semantic Explorer

Limitations & Outlook

- Semantic Explorer code release
  - Currently tied to IDA Pro -> support other frameworks (radare, …?)
- Improvements to graph exploration / referencing
  - Duplicate reduction
  - Fix occasional recursions
- Improvements to backtracking / dataflow analysis
  - Infer more calculated / constant arguments
  - Resolving more enums
- Expansion of set of semantic signatures
  - Contributions welcome! :)
  - Adoption of MITRE MAEC standard?
- Export / rendering of results

- IDAscope repository:
  - https://bitbucket.org/daniel_plohmann/simplifire.idascope