ReViver: Trace-Free Memory Data Forensics Through Speculative Symbolic Execution

Pengfei Sun

4N6 Research Group

Rutgers University

- 300+ benign processes
  - misbehaving
  - root-cause
  - pre-unknown vulnerabilities

ReViver's high level architecture

Alice

2. Memory allocation function hooking

5. Target process memory capture

1. Reviver client for Installation

3. System information (library versions, etc.)

6. Anomalous application's memory and executable

10. Memory data structure layout information

Cloud (ReViver)

4. Library source code parsing for data structure definition

7. (Past) Enhance the data structure database given the suspended execution state

8. (Present) Scan the memory dump for data structures and perform field-based analysis

9. (Future) Speculative future execution investigation for possible memory accesses

Past: Statistical Information

Goal:

- Collecting the statistical information about the executable's use of data structures

How:

- Best-Effort Partial Symbolic Execution
- Prior Knowledge Collection
- Library Parsing for Structure Definitions

Present: Static Heap Analysis

Goal:

- Reversing data type fine-grained for captured memdump

How:

- Trace-free heap reverse engineering
- Pruning candidate library structures
- Dump-library structure matching

Future: Speculative Forensics

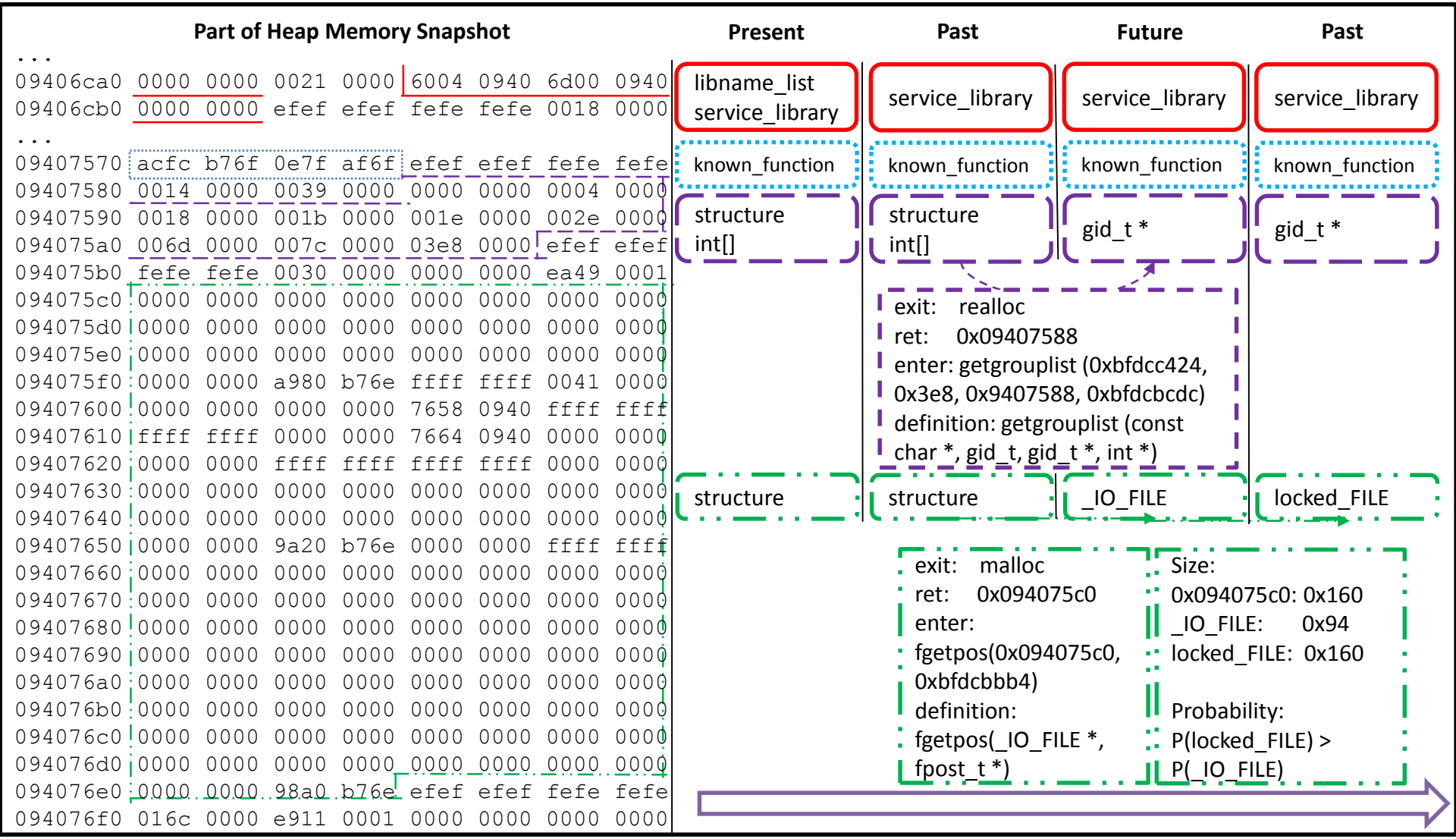
Goal:

- Confirm or correct data structure by type sinks

How:

- Revived speculative symbolic execution
- Type propagation for type forensics

CoreUtils Groups Case Study



Memory Data Forensics on Smart Grid SCADA System

- Learn about the memory layout and what data are available in memory
- Improve incident response
- Detect malicious modifications
- Figure out the root-cause of misbehaving

Q&A