Memory Based Dynamic Malware Analysis

Endre Bangerter, Jonas Wagner
Benjamin Urech, Patrick Schläpfer

Security Engineering Lab
Bern University of Applied Sciences
Memory forensics

Investigation

State Diffing

Memory tracing
Memory tracing

- Comprehensive capture of full system behavior, based on memory introspection

- May capture transient memory contents (i.e., short lived data & code)

- Hard to evade, reconstruction of system states from memory

- Novel techniques and algorithms to conduct dynamic malware analysis
System perspective

1. Malware → Memory sampling → Memory acquisition → Memory trace & meta data → Analysis

2.
Memory acquisition

- Based on virtual machine introspection, e.g. hook KVM core functionalities
- Trigger a new snapshot on certain guest events, e.g. system calls
- Good performance, e.g. 100 snaps/sec while having an interactive VM
- Operating system independence and stealth
Analysis

- Bridge the semantic gap
- Type 1: Data structure diffing
- Type 2: Content inspection
- Type 3: Temporal content inspection
Type 1: Data structure diffing - Code injections

explorer.exe

Module

Heap

Code

Time
Type 2: Content inspection - Code injection behavior

explorer.exe

dr-evil.exe

Heap

Code

Time
Type 3: Temporal content inspection - Self-modifying code

1                       2                        3                      4

Module

Code

Heap

Module

1  2  3  4
Type 1-3: Detecting code phases

- Characterize code phases: packer, loader, payload, etc. based on their lifetime in memory
- Detection of points in time where code is stable
- Remove redundancies through pattern matching
Finding correlations between samples
Finding correlations between samples

Payload: 20%

Loader: 90%
Conclusion

- Novel memory acquisition technique
- Memory inspection based analysis
- Automate some aspects of reverse engineering
- Provide entry points for further analysis
- Delivers relevant artifacts for malware correlations
Questions?

https://sel.bfh.ch

- 

jonas.wagner@bfh.ch