FAROS: Illuminating In-Memory Injection Attacks via Provenance-based Whole System Dynamic Information Flow Tracking

Meisam Navaki Arefi, Geoffrey Alexander,
Hooman Rokham, Aokun Chen, Michalis Faloutsos, Xuetao Wei,
Daniela Seabra Oliveira and Jedidiah R. Crandall

THE UNIVERSITY OF NEW MEXICO

UNIVERSITY of FLORIDA

UC RIVERSIDE

University of CINCINNATI

Problem

- In-memory Injection attacks.

- They are becoming more and more common.

- We built a **reverse engineering tool** to **flag** them and **give analysts the information they need to reverse engineer such malware.**

In-Memory Injection Attack

- Operates only on memory
- Acts very stealthy
- Hard to detect

Threat Model

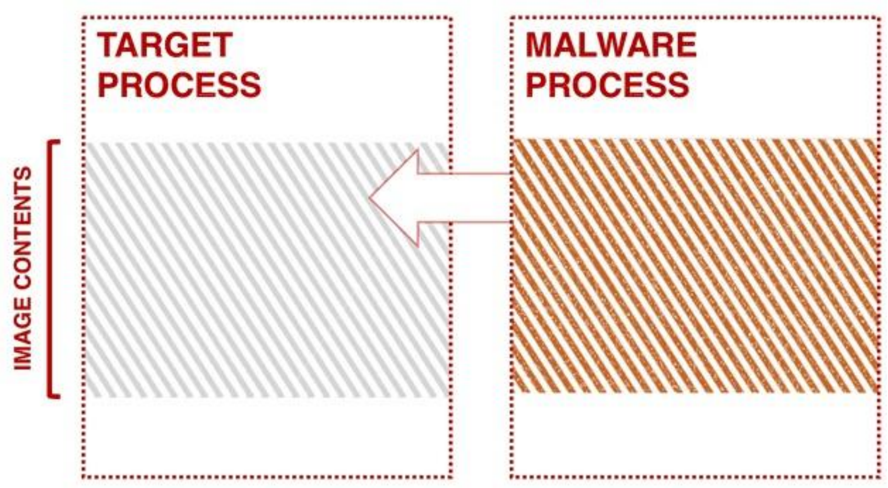
- Reflective DLL injection
- Process hollowing/replacement
- Code/process injection

Threat Model - Reflective DLL Injection

- **Reflective DLL injection** refers to loading a DLL from **memory** rather than from disk.
- Windows doesn’t have such loading function.
- Write your own load function: Omitting some of the things Windows normally does, e.g. registering the DLL as a loaded module.

Threat Model - Process Hollowing

- Start a process in a suspended state.
- Replace the process image with a malicious one.
- Run the process.
- Easy!



Threat Model - Code Injection

- Write the malicious code directly to the address space of the target process.
- Have the target process run the code.
- Easy!

Motivation

- Current malware analysis solutions, e.g. CuckooBox and memory forensics tools, are no match.
- An analyst needs visibility into memory throughout the execution to flag such attacks.
- **Question:**
    - **How the attack was conducted?**
    - **What is the source of the attack?**
    - **...**

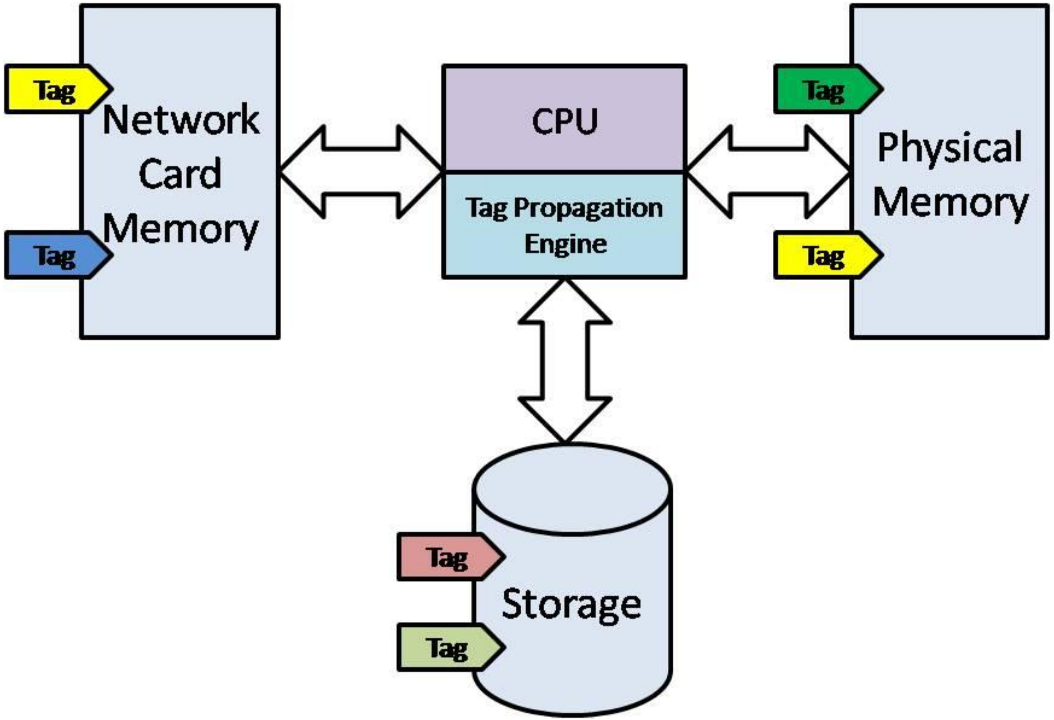
Dynamic Information Flow Tracking (DIFT)

- Makes systems transparent for attack detection, enforcement of security policies and forensics*

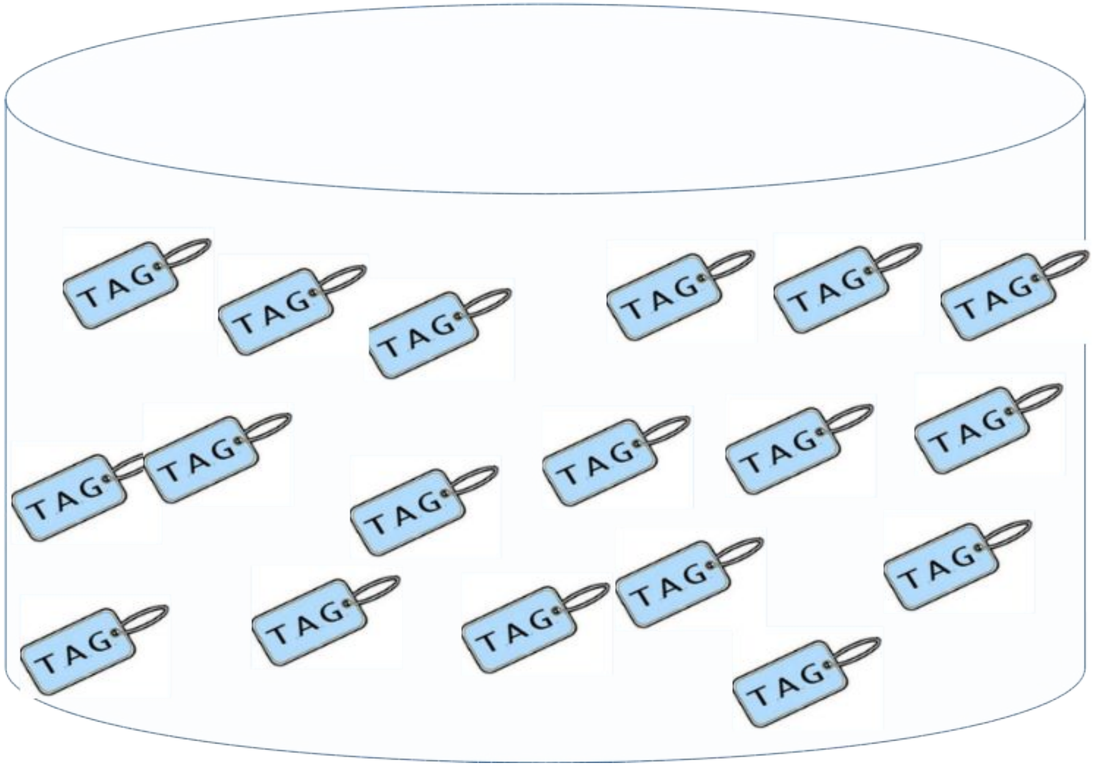
*Suh et al. 2004, Minos (Crandall and Chong 2004), TaintCheck (Newsome and Song 2005), and Vigilante (Costa et al. 2004)

DIFT - How?

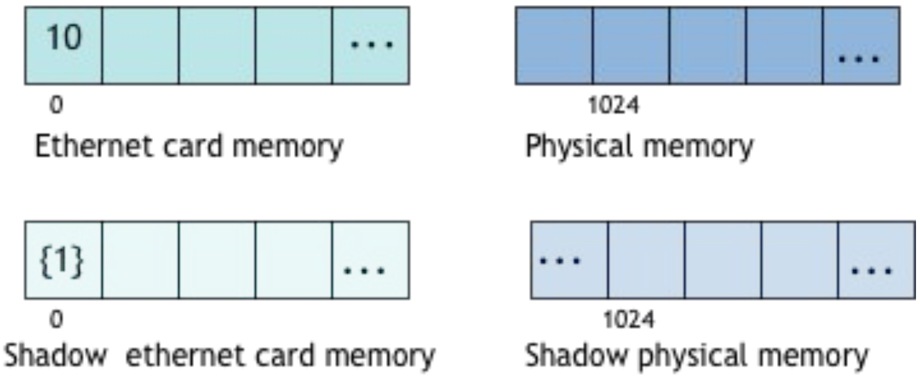
I. Introduce the tags/taints
II. Propagate the tags
III. Check the status of tags



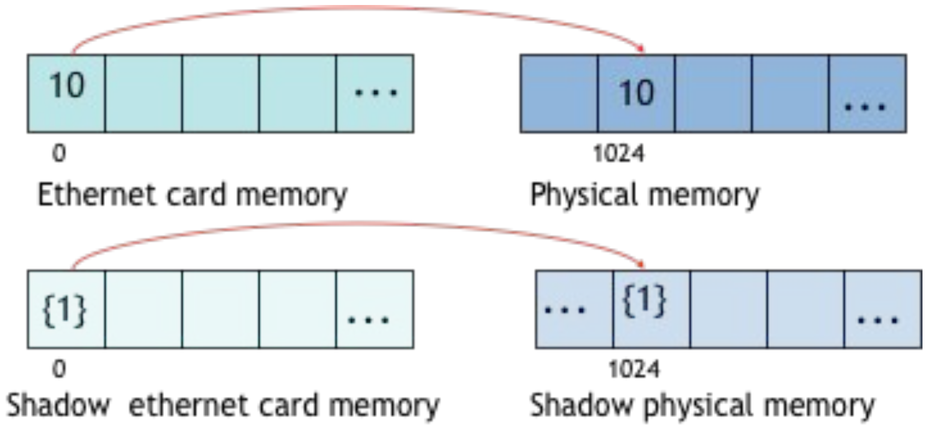
Shadow Memory



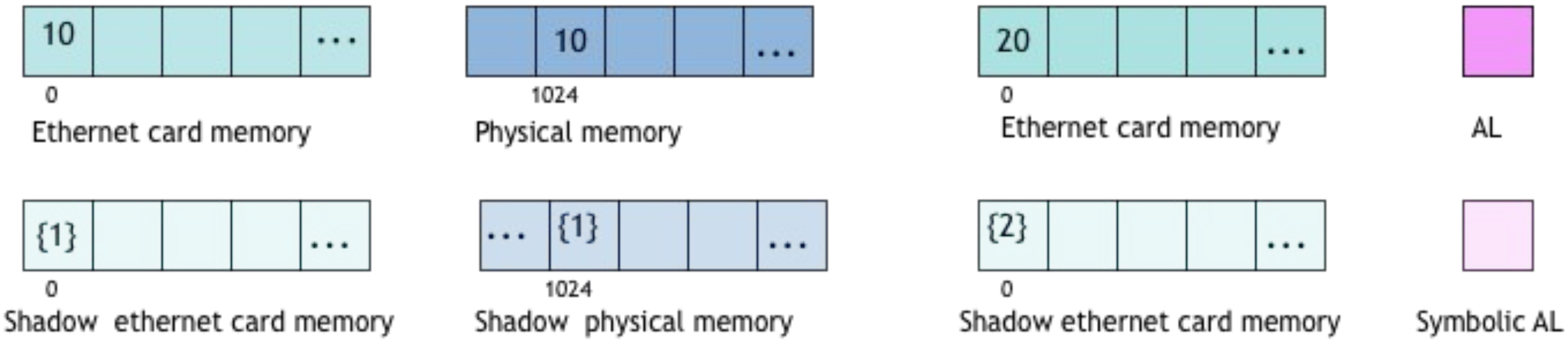
DIFT Example



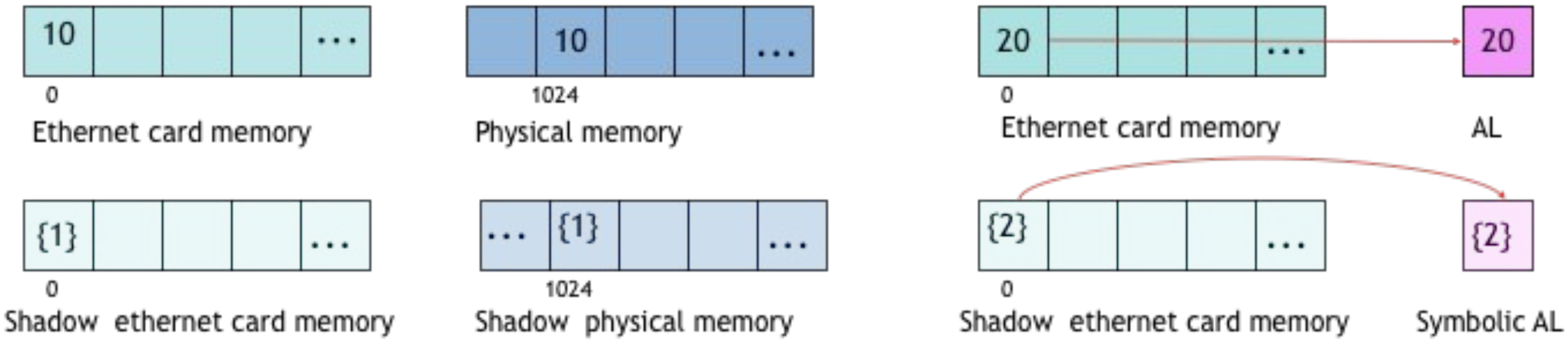
DIFT Example



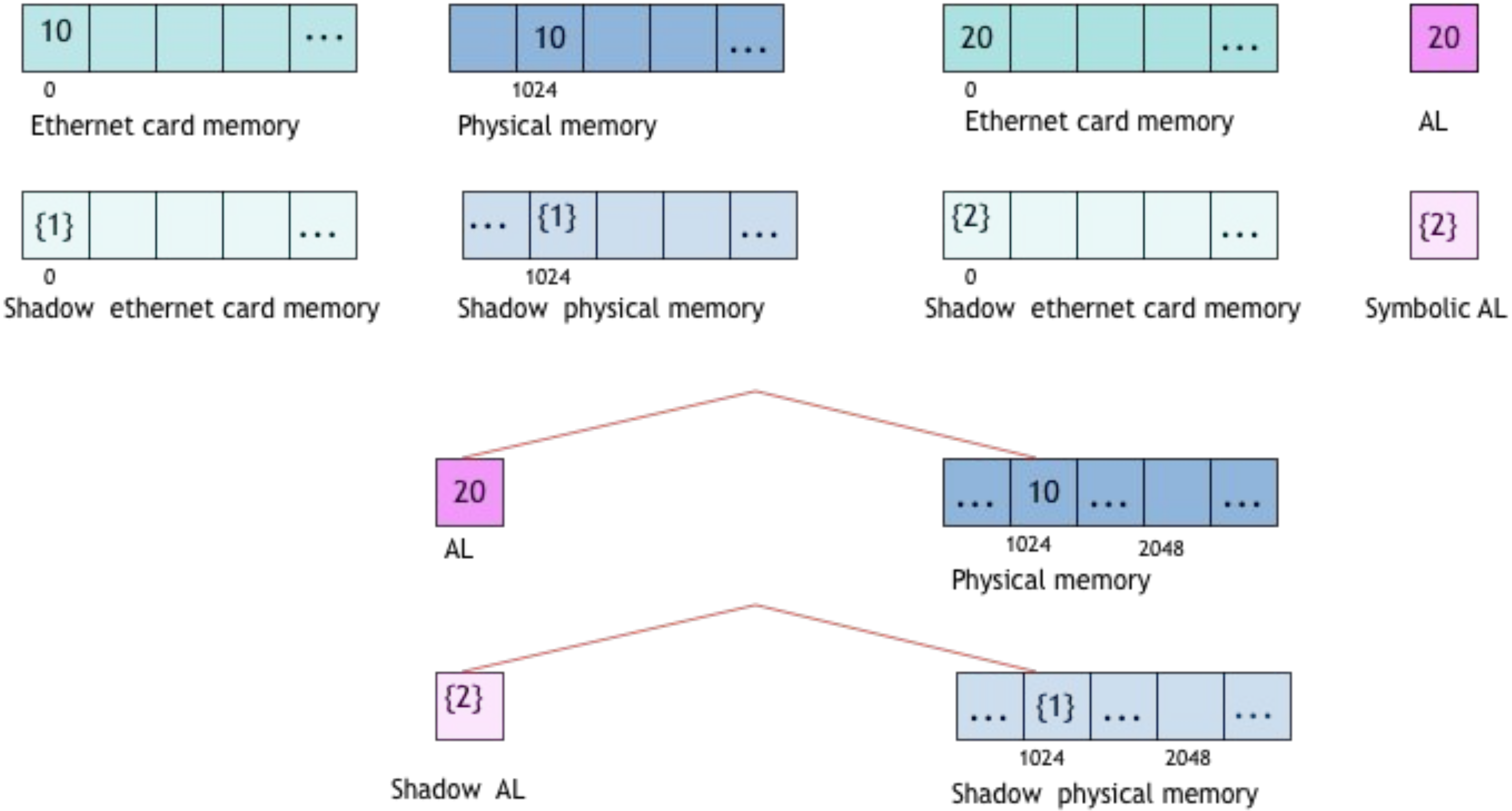
DIFT Example



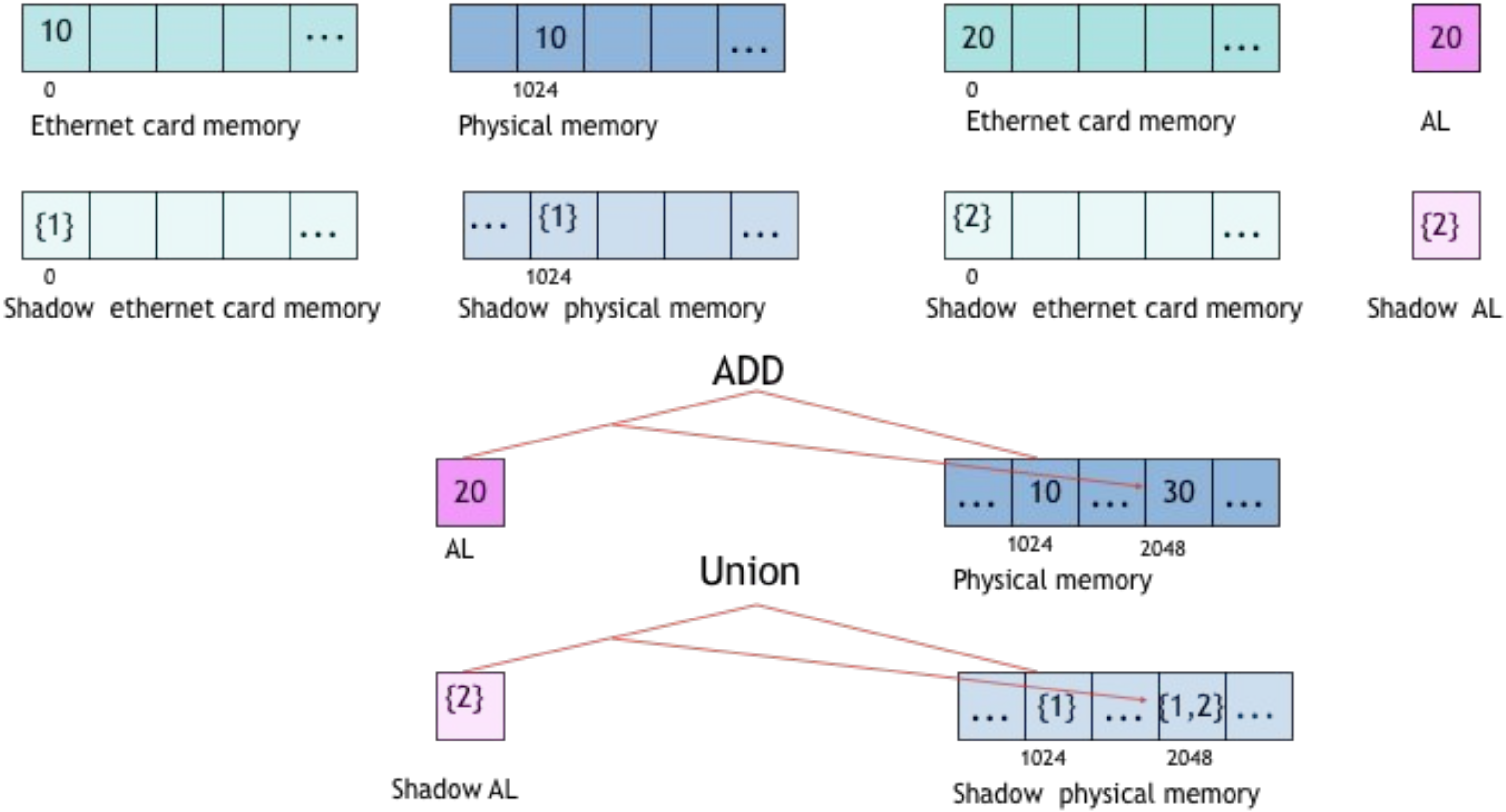
DIFT Example



DIFT Example

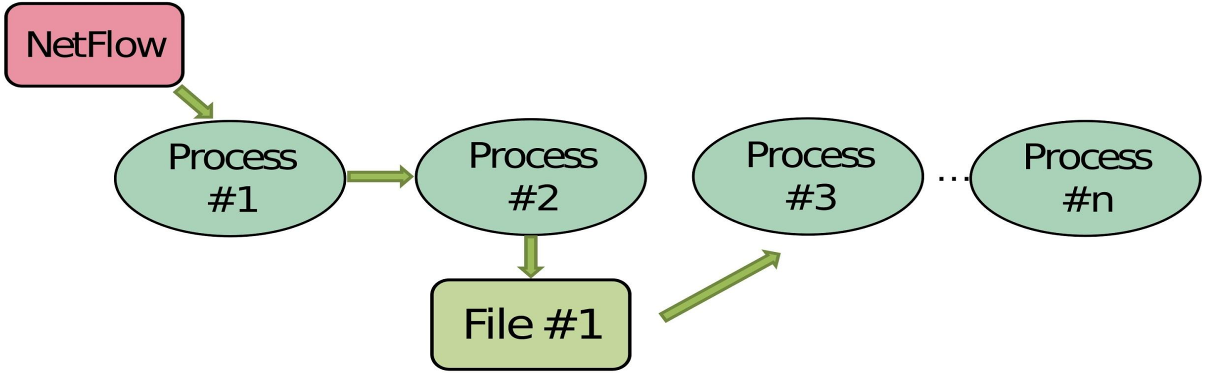


DIFT Example



Provenance List

- Each byte could have a list of tags (provenance list).



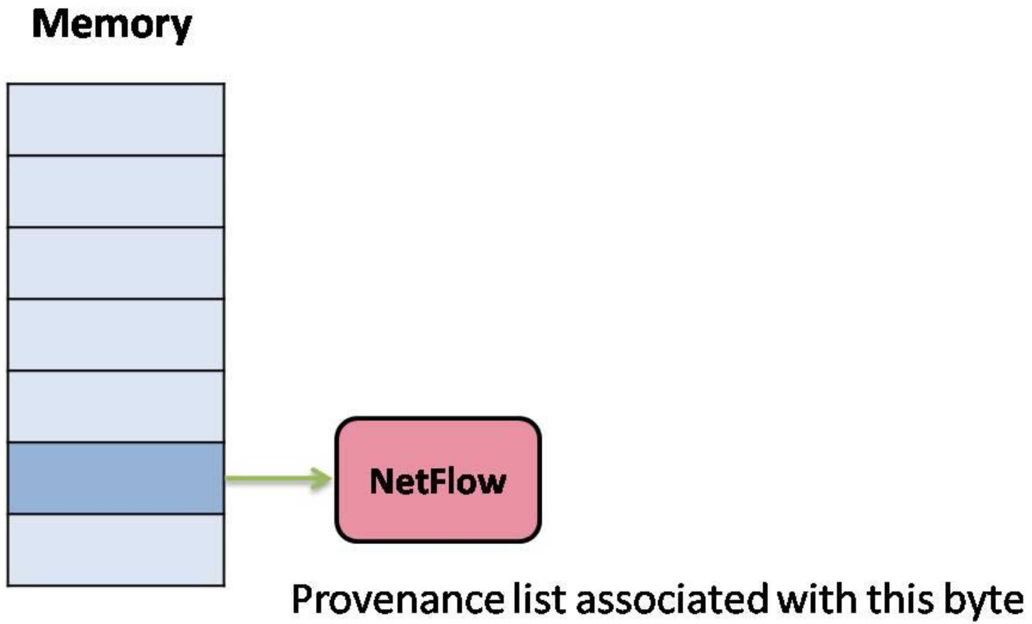
A provenance list for a specific byte

Tag Confluence

- Two or more tags of different types can “come together”.

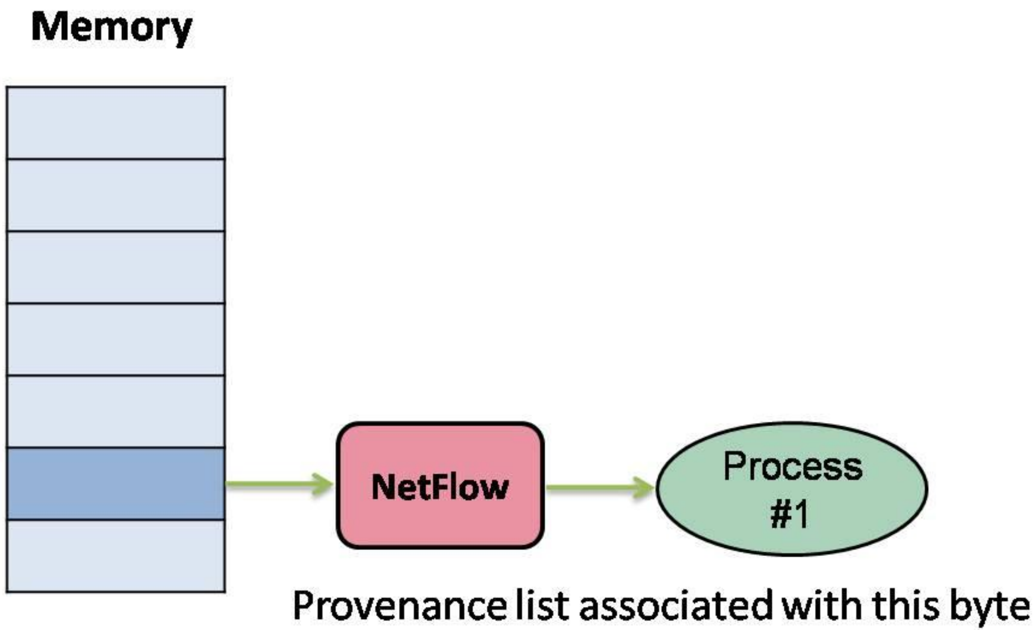
Tag Confluence

- A bytes comes in from the network and then moves to the physical memory.



Tag Confluence

- Process #1 accesses that byte.



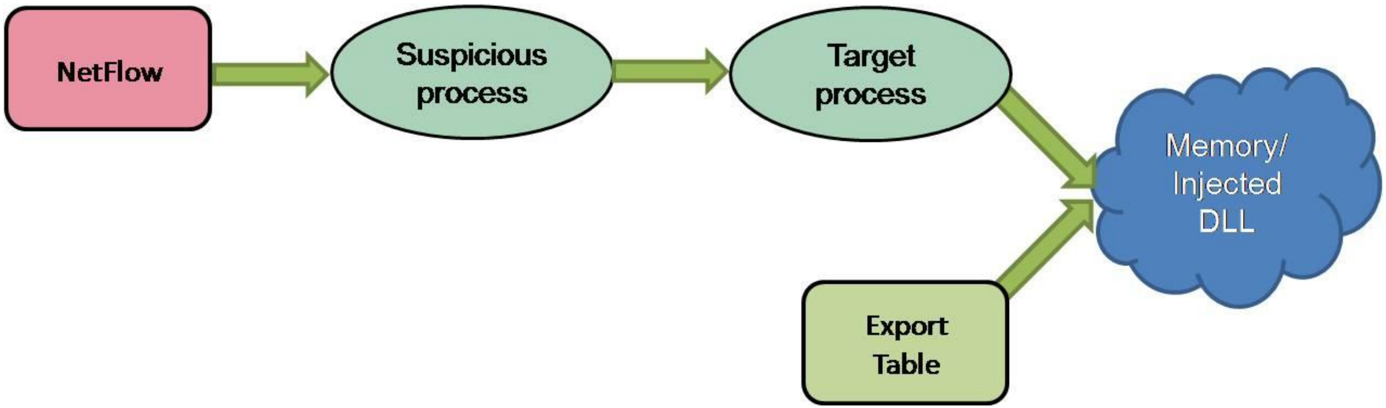
Flagging Policy via Provenance-based DIFT

Data coming in from the network (**Netflow tag**) **SHOULD NOT** "come together" with linking/loading data exported by the kernel (**export table tag**).

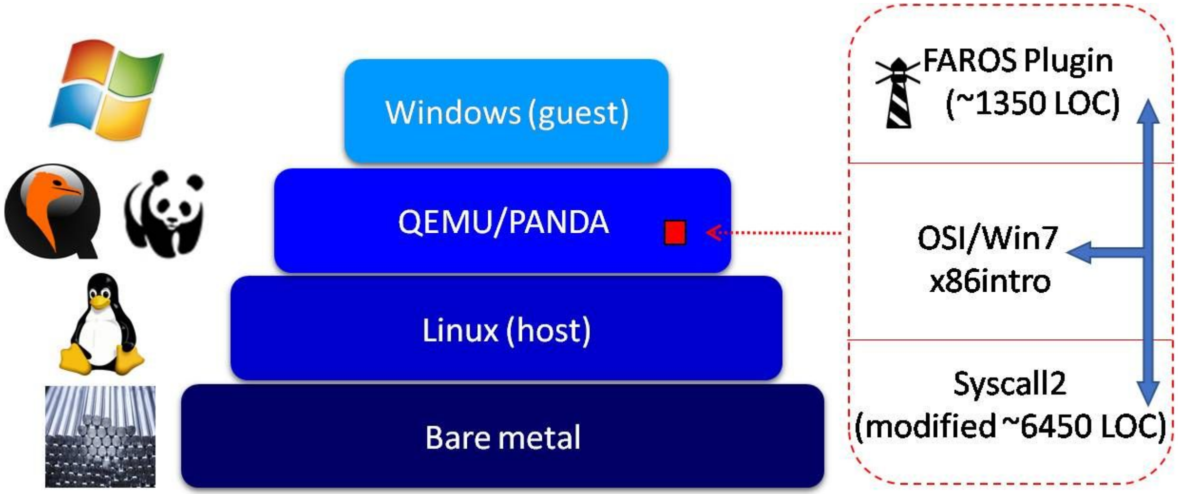
That shouldn't happen under normal circumstances!

Flagging Policy via Provenance-based DIFT

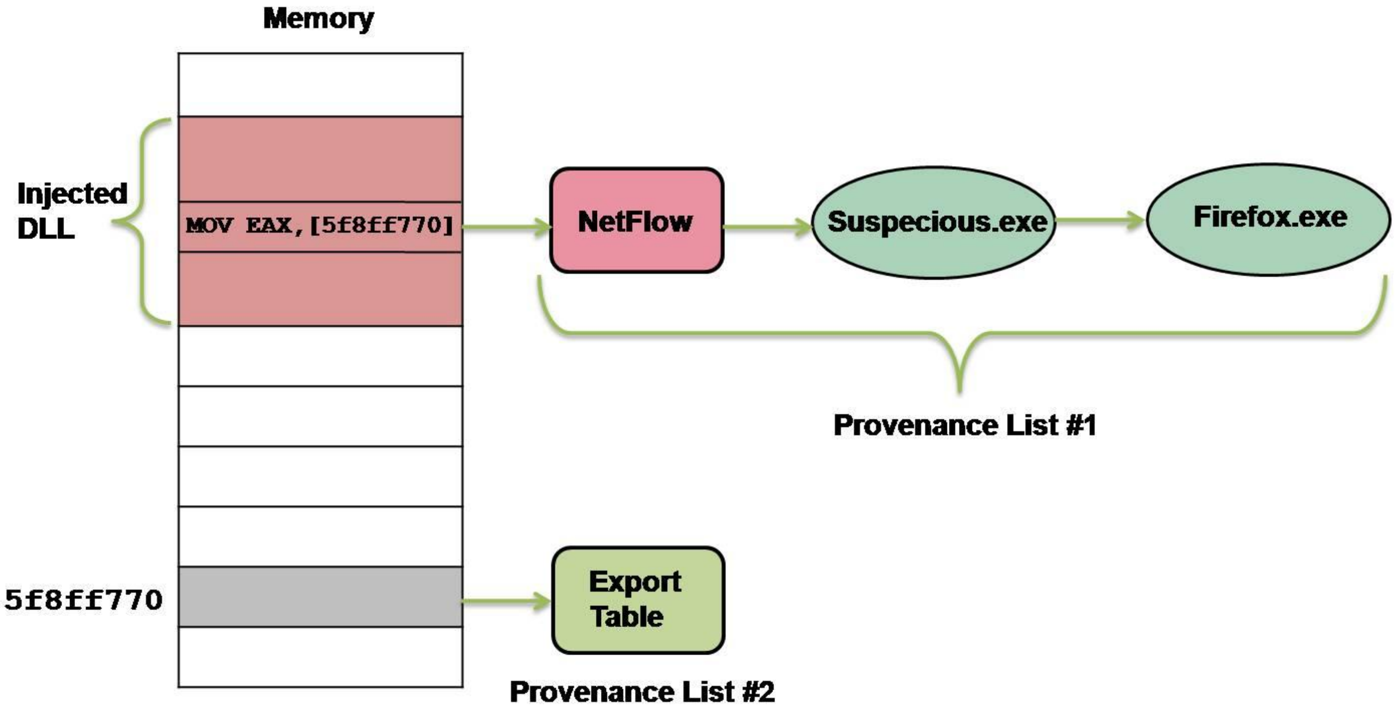
- **Tag confluence heuristic:**



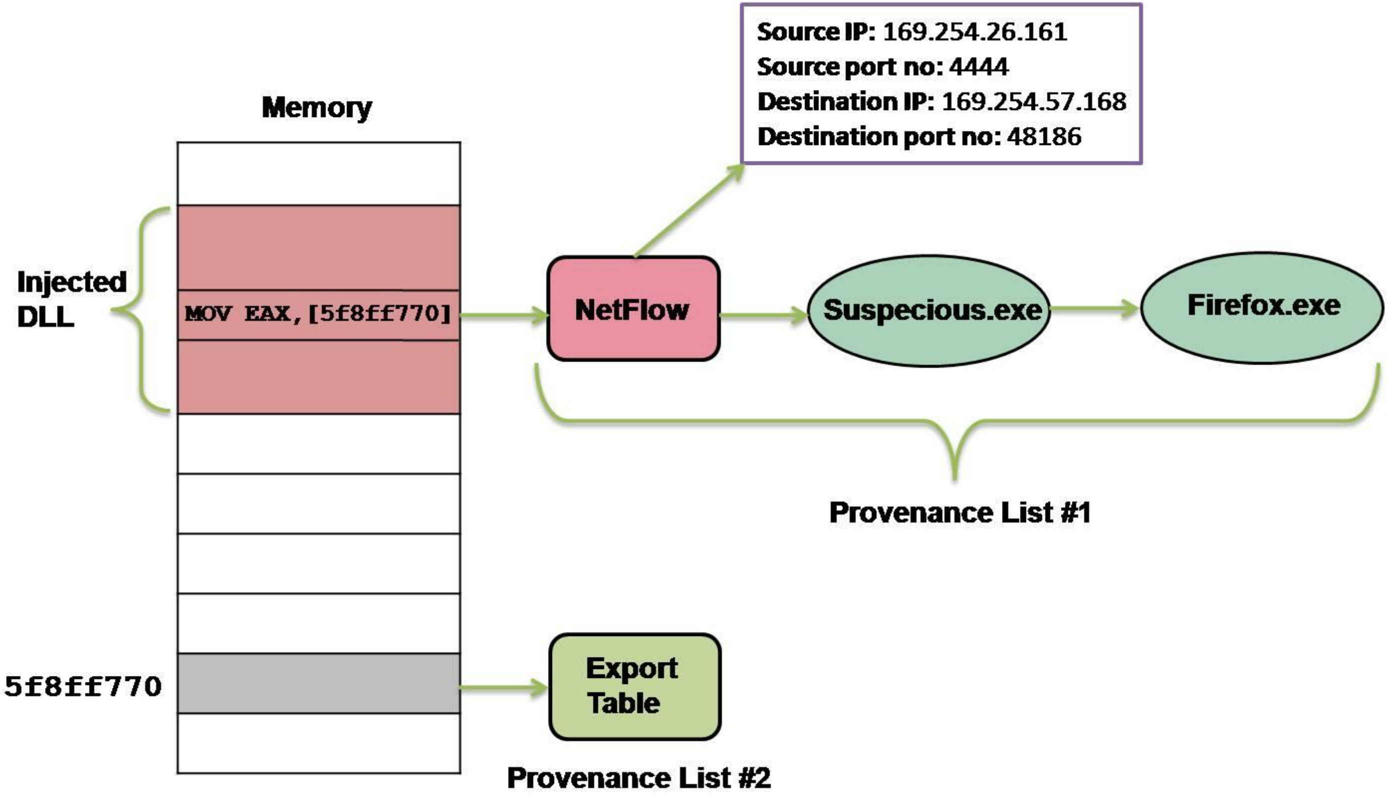
System Architecture



Results - Reflective DLL Injection



Results - Reflective DLL Injection



Comparison with CuckooBox

- Most popular open-source malware analysis system.

- We tested CuckooBox on in-memory injection attacks.

- CuckooBox (along with *malfind* and *Volatility* plugins) provided limited visibility into these attacks.

- **With CuckooBox, we are blind as to how the attack was conducted.**

True/False Positive Analysis

- Tested against **6** memory injection attacks and successfully flagged them all.

- Tested against **90** non-injecting malware samples and **14** benign software from various categories.
    - FAROS presented a very low false positive rate of **2%**.

Performance Evaluation

- Performance is not a priority for FAROS.
- Focused on providing a low false positive rate.
- FAROS’ slowdown is **56X** compared to QEMU.

Conclusions

- Presented FAROS, a DIFT-based reverse engineering tool, which can illuminate in-memory injection attacks.
- Tag confluence as a promising heuristic.
- Very low false positive (2%).
- FAROS
  - can save reverse engineers substantial time and effort in practice.
  - can provide reverse engineers with valuable information about any in-memory injection attacks.
- FAROS is open source:
  - https://github.com/mnavaki/FAROS

Acknowledgments

- Our reviewers and our shepherd, Etienne Riviere.

- DARPA Trusted Computing Project (Grant No. FA8650-15-C-7565) and the U.S. National Science Foundation (Grant Nos. #1518523, #1518878).

- **NSF disclaimer:** Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

Thank you!

mnavaki@unm.edu