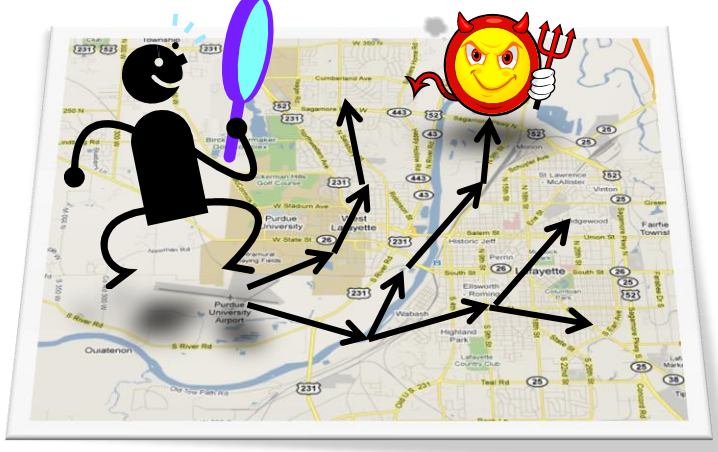


# CERIAS

the center for education and research in information assurance and security

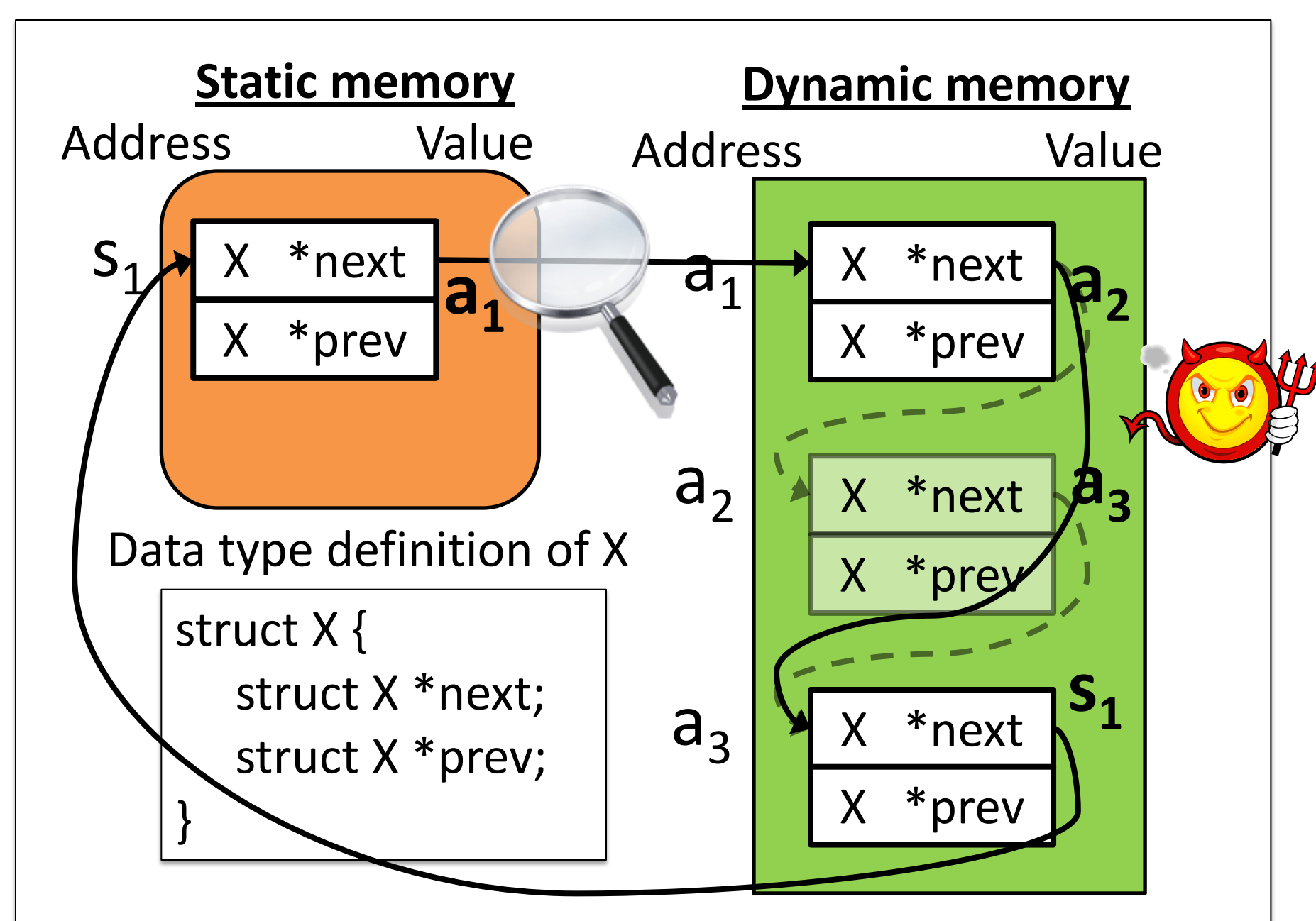
## LiveDM: Kernel Malware Analysis with Un-tampered and Temporal Views of Dynamic Kernel Memory

Junghwan Rhee\*, Ryan Riley+, Dongyan Xu\*, Xuxian Jiang‡  
 \*Purdue University and CERIAS, +Qatar University, ‡NCSU



### State-of-the-art Memory Mapping

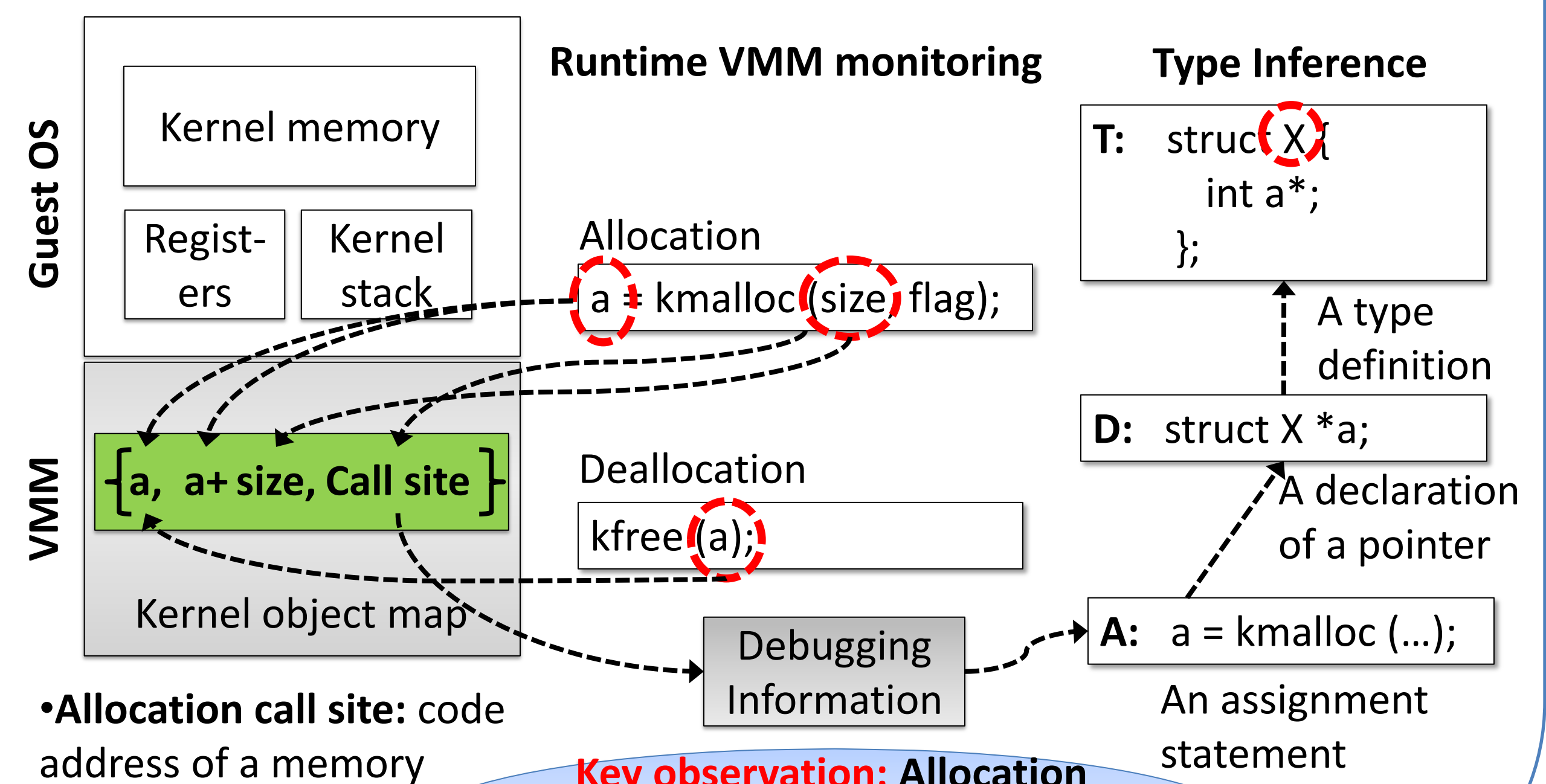
- Kernel object maps are built by recursively traversing pointer connections starting from static objects. (Type-projection Mapping)
- Maps are subject to pointer manipulation.
- Asynchronous due to its base on memory snapshots



Kernel memory view is subject to malware manipulation.

### Allocation-driven Mapping Approach

- Kernel objects are identified by transparently capturing kernel memory function calls.
- Memory ranges are extracted from function arguments and return values.
- Call stack information is used to derive data types.



• Allocation call site: code address of a memory allocation call

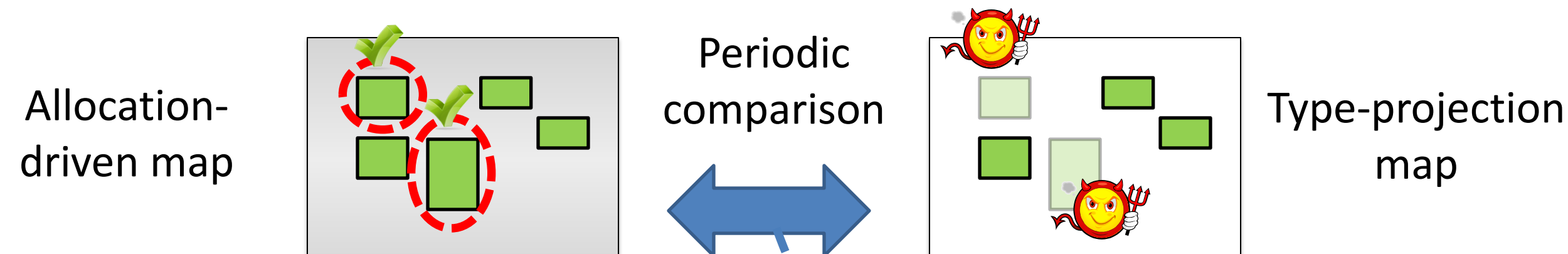
Key observation: Allocation call site can be used to infer the object's type.

LiveDM: Live Dynamic Kernel Memory Map

### Applications of Allocation-driven Mapping

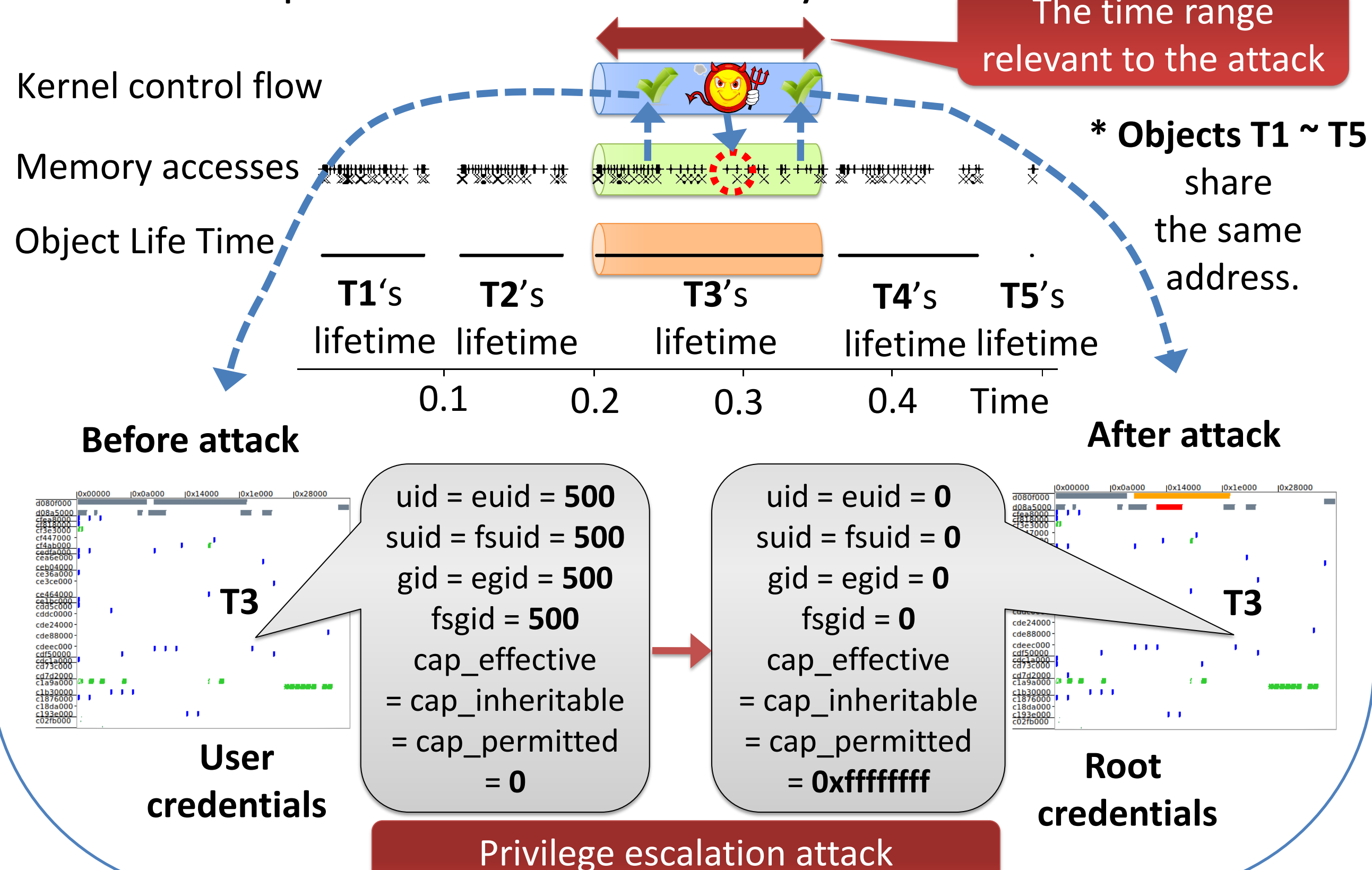
#### • Un-tampered view

Systematic detection of kernel data hiding attacks



#### • Temporal view

Temporal Kernel Rootkit Analysis



### Detection of Rootkit Attacks Hiding Kernel Objects

Rootkit Name	# of Hidden Objects	Manipulated Data		Attack Vector
		Type	Field	
hide_lkm	# of hidden drivers	module	next	/dev/kmem
fuuld	# of hidden processes	task_struct	next_task, prev_task	/dev/kmem
cleaner	# of hidden drivers	module	next	LKM
modhide	# of hidden drivers	module	next	LKM
hp	# of hidden processes	task_struct	next_task, prev_task	LKM
linuxfu	# of hidden processes	task_struct	next_task, prev_task	LKM
modhide1	1	module	next	LKM
kis 0.9	1	module	next	LKM
adore-ng 2.6	1	module	list.next, list.prev	LKM
ENYELKM	1	module	list.next, list.prev	LKM



Demo

Slides

Paper

Author