Multi-Aspect Profiling of Kernel Rootkit Behavior

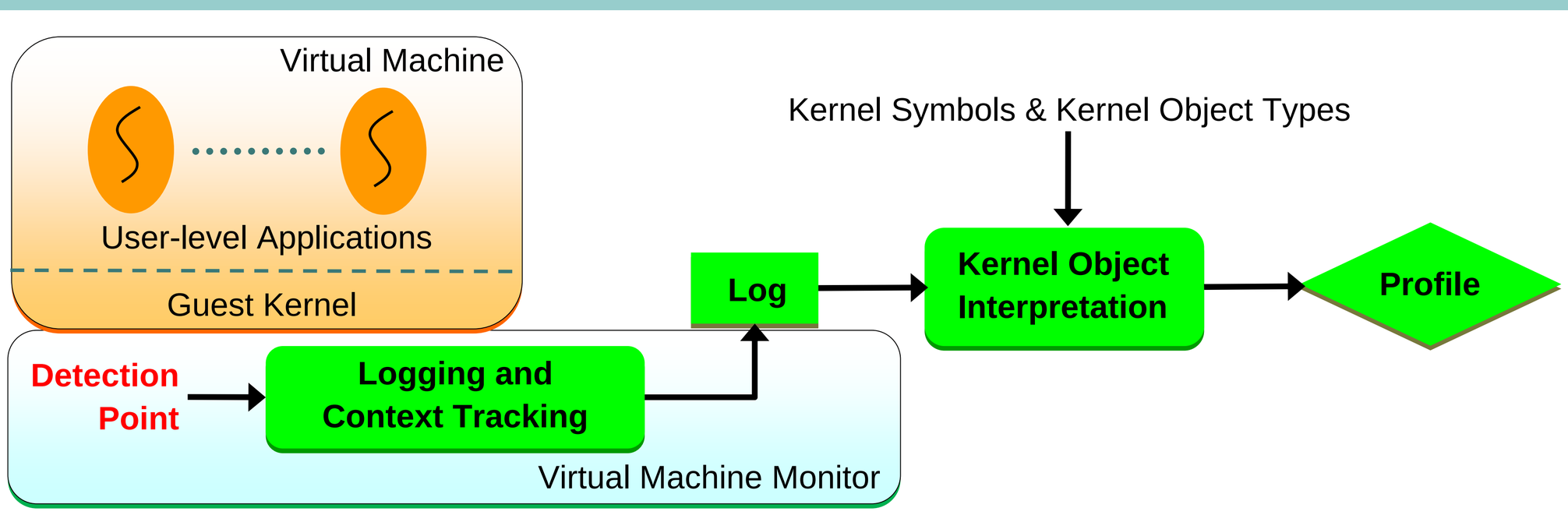
Ryan Riley[1], Xuxian Jiang[2], Dongyan Xu[1]

[1] Purdue University, [2] North Carolina State University

1. Problem

Profiling a kernel rootkit's behavior in a "live" system (e.g., honeypot) in multiple aspects: (1) *hooking behavior*, (2) *manipulated kernel objects*, (3) *affected system calls*, and (4) *malicious kernel code executed*.

2. PoKeR: A Profiler of Kernel Rootkits



3. Challenges and Solutions

- Real-time switch to kernel rootkit profiling mode → Adapting instantaneous rootkit detection system (e.g., *NICKLE*)
- Reverse VMI: given a memory location, infer the kernel object → Combat tracking: following the reads/writes of kernel rootkits
- Association of rootkit code execution with system calls → System call introspection and process-based association

4. Results

- 10 real-world kernel rootkits profiled
- High accuracy compared with manual analysis
- Telling researcher *what* a rootkit did and helping her determine *why*

Name	Code	Kernel Objects Modified	Syscalls Affected	Attack Type
Adore 0.42	770 Instr.	`sys_call_table[2,4,5,6,18,37,39,84,106]` `sys_call_table[107,120,141,195,196,220]`	2 - fork, 4 - write, 5 - open, 6 - close, 195 - stat64, 196 - lstat64, 220 - getdents64	syscall hook

Adore 0.53	733 Instr.	`sys_call_table[1,2,6,26,37,39,120,141,220]`
		`proc_net->subdir->next->(...)->next->get_info`
		`proc_root_inode_operations->lookup`
			1 - exit, 2 - fork,
			3 - read, 5 - open,
			6 - close,
			85 - readlink,
			195 - stat64,
			220 - getdents64
				syscall hook,
				data hook

Adore-ng 0.56	785 Instr.	`proc_net->subdir->next->(...)->next->get_info`
		`proc_root_inode_operations->lookup`
		`proc_root_operations->readdir`
		`ext3_dir_operations->readdir`
		`ext3_file_operations->write`
		`unix_dgram_ops->recvmsg`
			3 - read
			5 - open
			85 - readlink
			195 - stat64
			220 - getdents64
				data hook

NC STATE UNIVERSITY

EUROSYS09 30TH MARCH TO 03RD APRIL 2009 IN NUREMBERG, GERMANY

PURDUE UNIVERSITY