Intrusion Detection in Virtual Machines

Alex Dehnert

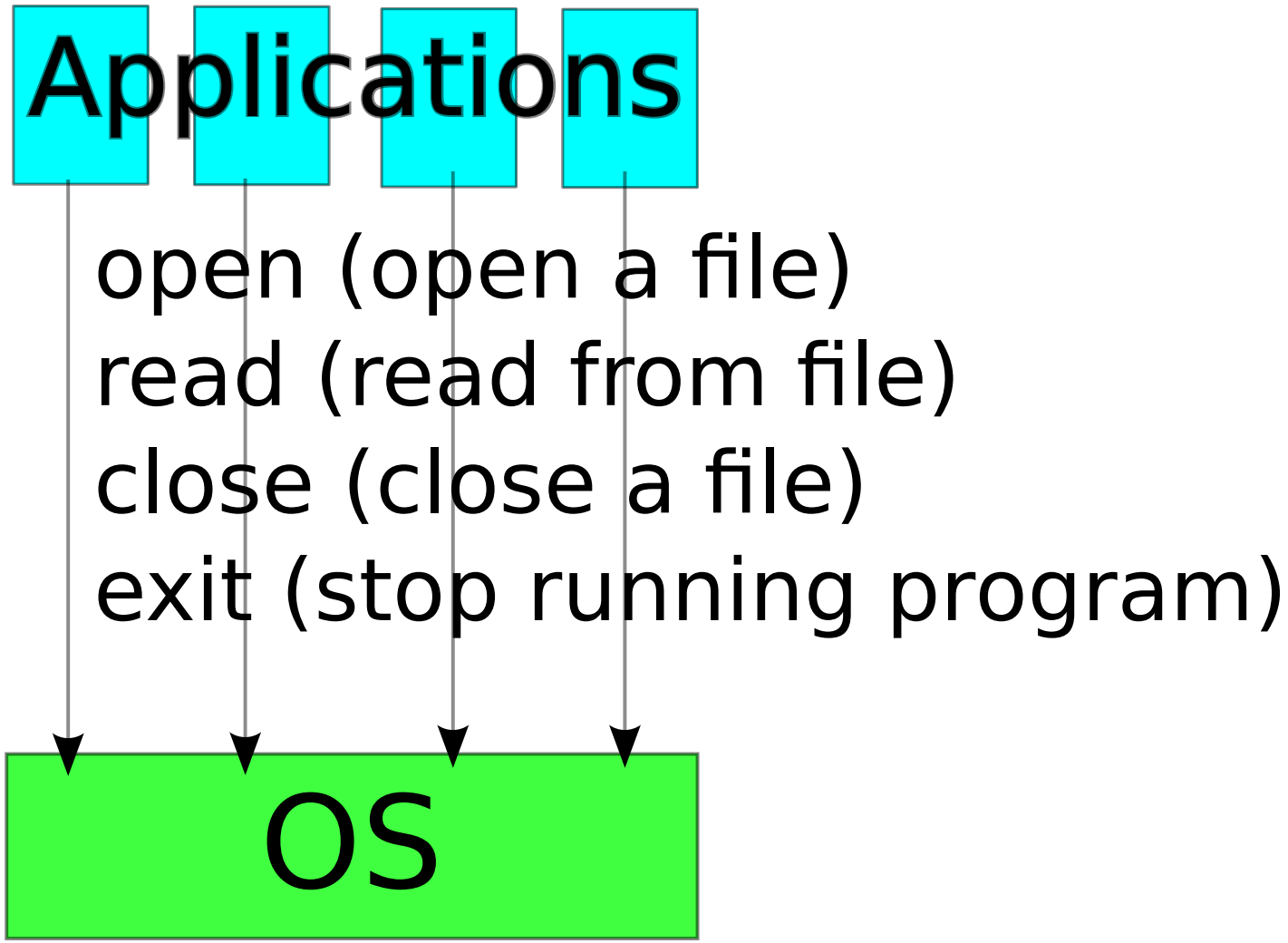
VI-A – VMware

Fall 2011

Problem

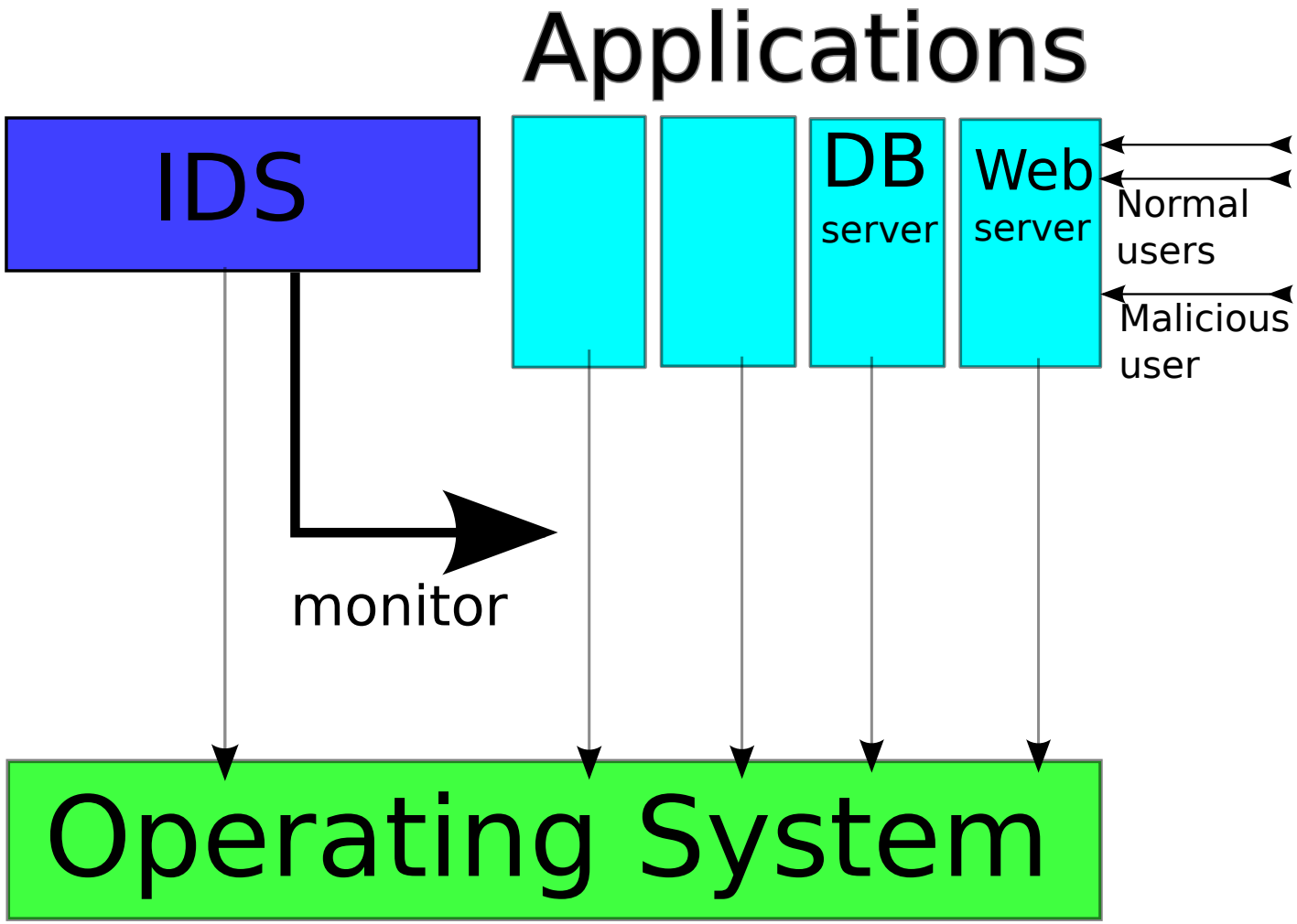
- Security is a growing issue
- Huge amounts of money are being spent, both to defend against attacks and to clean up after them
- Host-based intrusion detection systems monitor servers to detect attacks
- Security software is sometimes specifically targeted

System calls (syscalls)

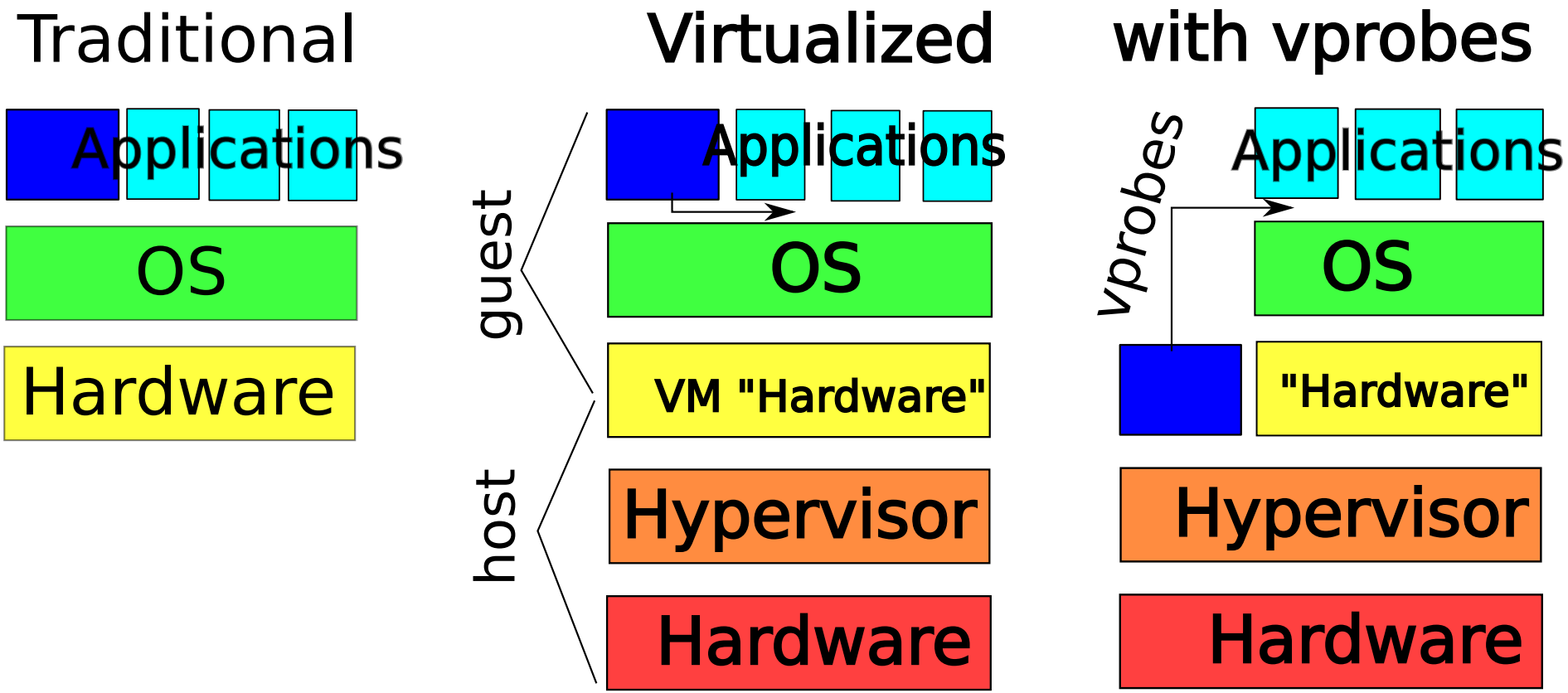


Intrusion detection

- Rich variety of work involving host-based intrusion detection
- Many based on patterns of system calls



How can virtual machines help?



Goal

Provide an intrusion detection system that is robust against attacks from malware, by leveraging vprobes to run the agent on the host

Prior Art

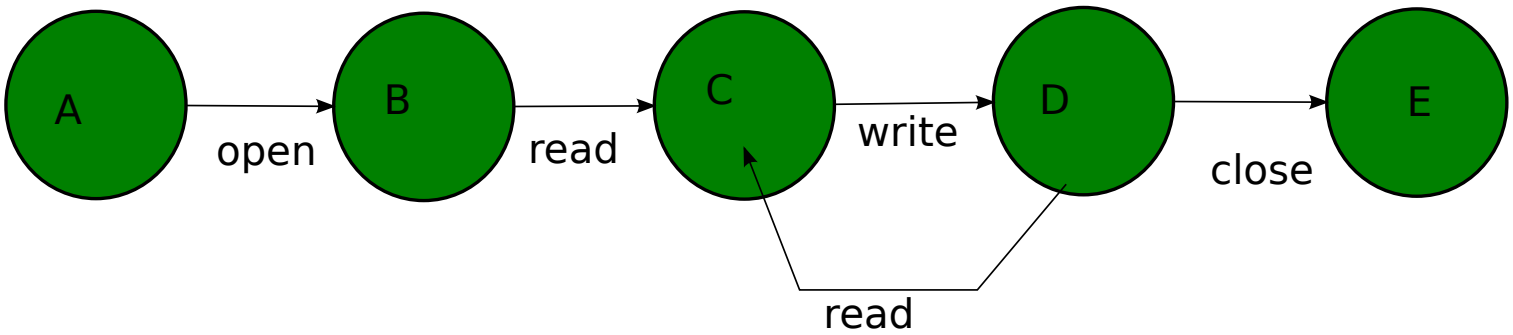
Much previous work on how to determine whether or not some sequence of syscalls is allowed

Prior Art: Custom rules

- Hand-written to match whatever the server in question is doing
- "Only allow the web server to access files under /var/www/"
- "Only allow the ssh server binary to bind to port 22"

Prior Art: Finite Automata

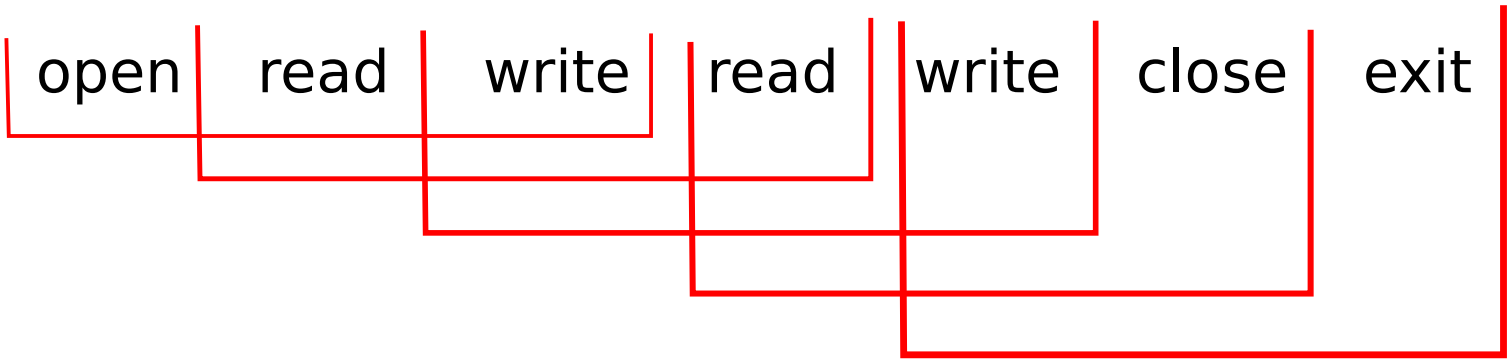
- Maintain current "state" associated with the syscalls so far (and other info)
- Define transitions between the various states
- When the appropriate syscall is seen, transition to the appropriate new state:



- If an unfamiliar syscall is made, alert

Prior Art: stide

- Sequence Time-Delay Embedding
- Sequences (*n*-grams) of syscalls



- Look at rolling window of twenty sequences at a time. If enough of them are unknown, alert.

What's new?

- Intrusion detection based on syscalls *isn't* — plenty of people have done that
- However, generally, that's vulnerable to attacks against the agent
- So, develop a version that gathers data using vprobes

Work required

- vprobe script to get syscall information from the kernel (Summer 2011)
- Build analysis modules (begun Summer 2011)
- Build up collection of exploits and normal behaviors to test against

Timeline

Week 1	Get set up at VMware again
Week 2	Re-familiarize with vprobes
Week 3	Find off-the-shelf exploits for Apache and other services
Week 4	Write a custom Apache exploit
Week 5	More exploit-writing
Week 6	Review literature on intrusion detection again; begin writing analyzer (e.g., finite automata)
Week 7	Continue literature-based analyzer
Week 8	Start custom analyzer (e.g., SVM)
Week 9	Continue custom analyzer
Week 10	Code cleanup and begin code review
Week 11	Finish code review; begin writing report
Week 12	Finish writing report

Risks

Big risk: schedule slippage Room to cut in both the exploit finding/writing and the analyzers

Can't come up with a novel analyzer Not key to the project. The main novelty is supposed to be in the vprobes component, anyway.

vprobes not powerful enough My team is the vprobes team. We can probably add required features.

Conclusion

- Security becoming increasingly important
- Host-based intrusion detection vulnerable to the agent being attacked
- Project: *Enhance effectiveness of intrusion detection systems by moving the vulnerable agent onto the host, and use vprobes to look inside the VM*
- Extensive literature exists on how to examine the data the agent/vprobes gather