Hprobes: Dynamic VM Monitoring

Z. J. Estrada, C. Pham,
Z. Kalbarczyk, R. K. Iyer

ACC Meeting 2014-09-24

Monitoring Techniques

- Passive

Monitoring Techniques

- Passive
  - Polling

Monitoring Techniques

- Passive
  - Polling
  - "Traditional"

Monitoring Techniques

- Passive
  - Polling
  - "Traditional"
- Active

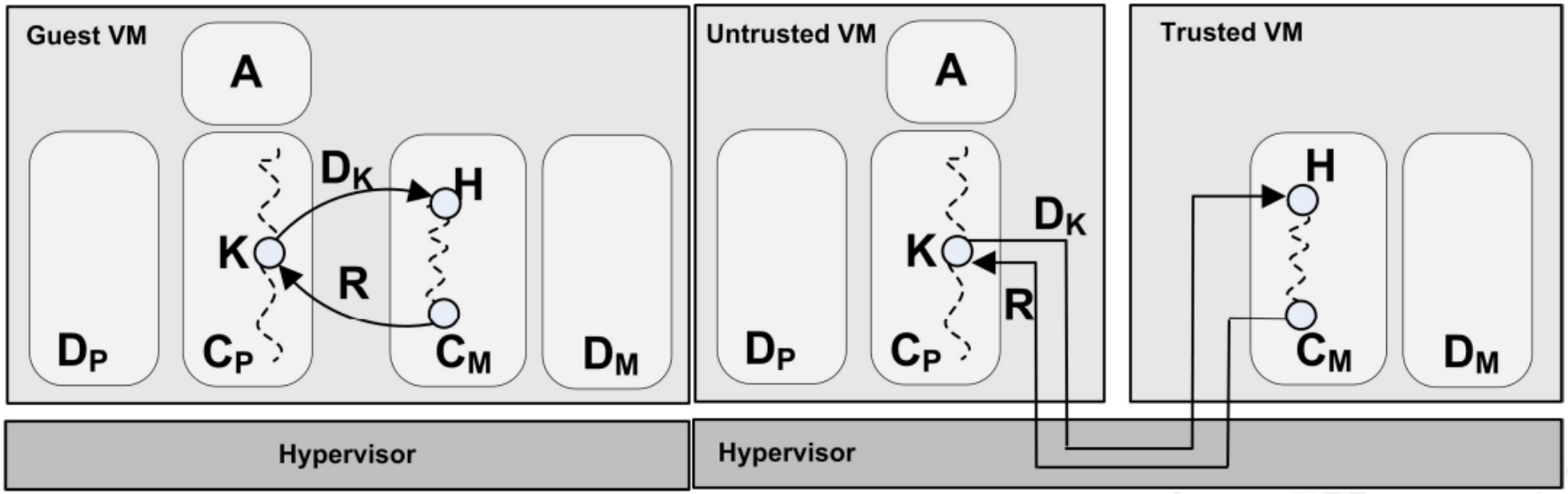
Monitoring Techniques

- Passive
  - Polling
  - "Traditional"
- Active
  - Event-driven

Monitoring Techniques

- Passive
  - Polling
  - "Traditional"
- Active
  - Event-driven
  - HyperTap (and others...)

Active Monitoring: Hooks



Hooks - Lares

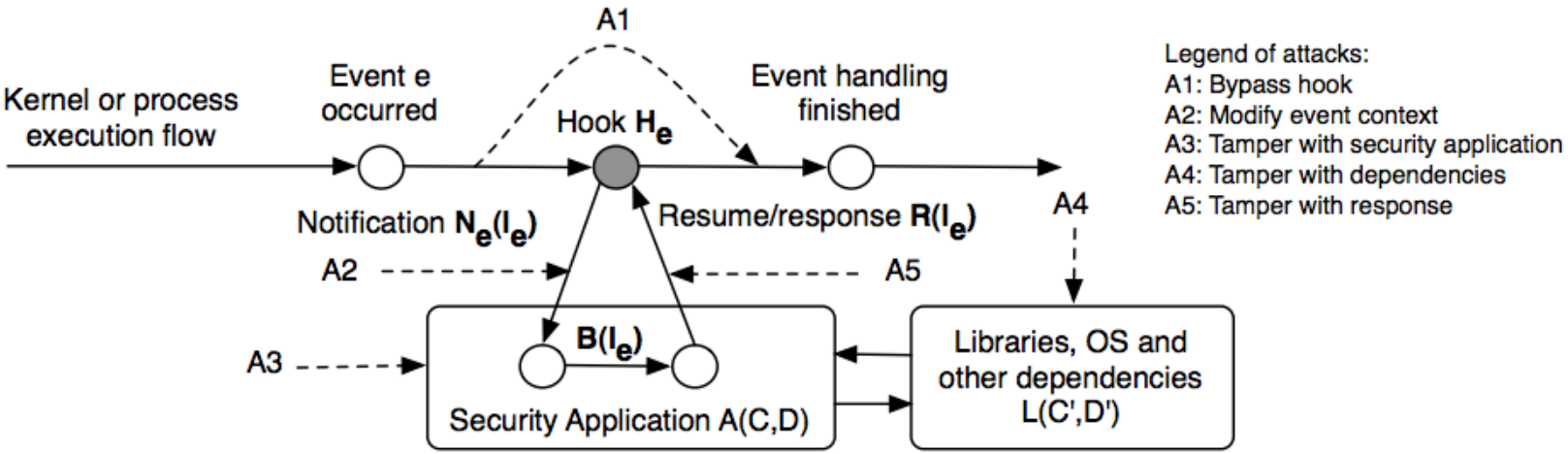
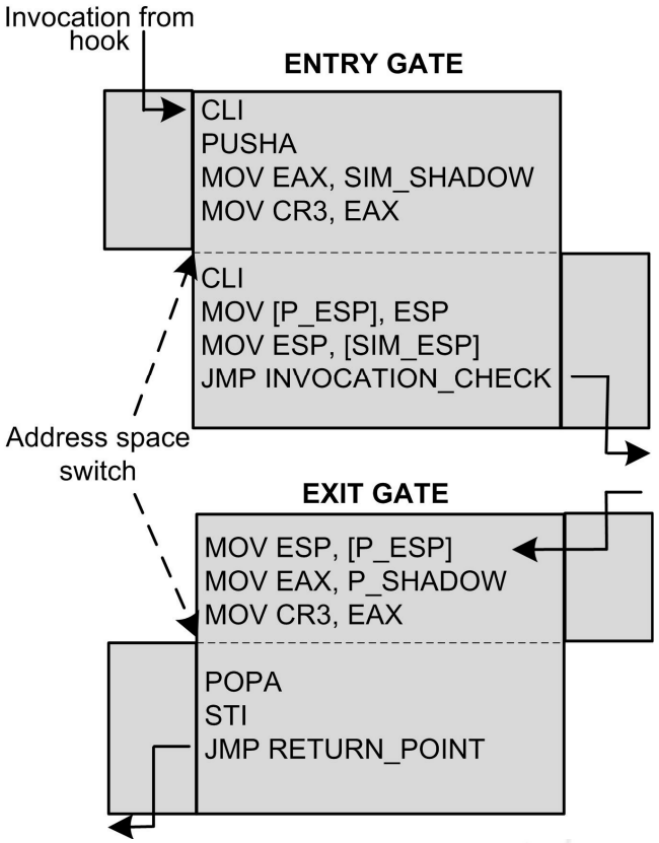


Figure 1: Formal model of secure active monitoring shown with potential attacks.

Payne, Bryan D., et al. "Lares: An architecture for secure active monitoring using virtualization." Security and Privacy, 2008. SP 2008. IEEE Symposium on. IEEE, 2008.

Hooks - SIM



Sharif, Monirul I., et al. "Secure in-vm monitoring using hardware virtualization." Proceedings of the 16th ACM conference on Computer a communications security (CCS). ACM, 2009.

Properties of Lares, SIM, ...

+ Active monitoring

Properties of Lares, SIM, ...

+ Active monitoring
+ Protected hooks

Properties of Lares, SIM, ...

+ Active monitoring
+ Protected hooks
− Not transparent - OS driver, etc...

Properties of Lares, SIM, ...

+ Active monitoring
+ Protected hooks
− Not transparent - OS driver, etc...
− Not dynamic - boot time config

Hprobes

- Active monitoring

Hprobes

- Active monitoring
- Transparent

Hprobes

- Active monitoring
- Transparent - OS agnostic

Hprobes

- Active monitoring
- Transparent - OS agnostic
- Dynamic

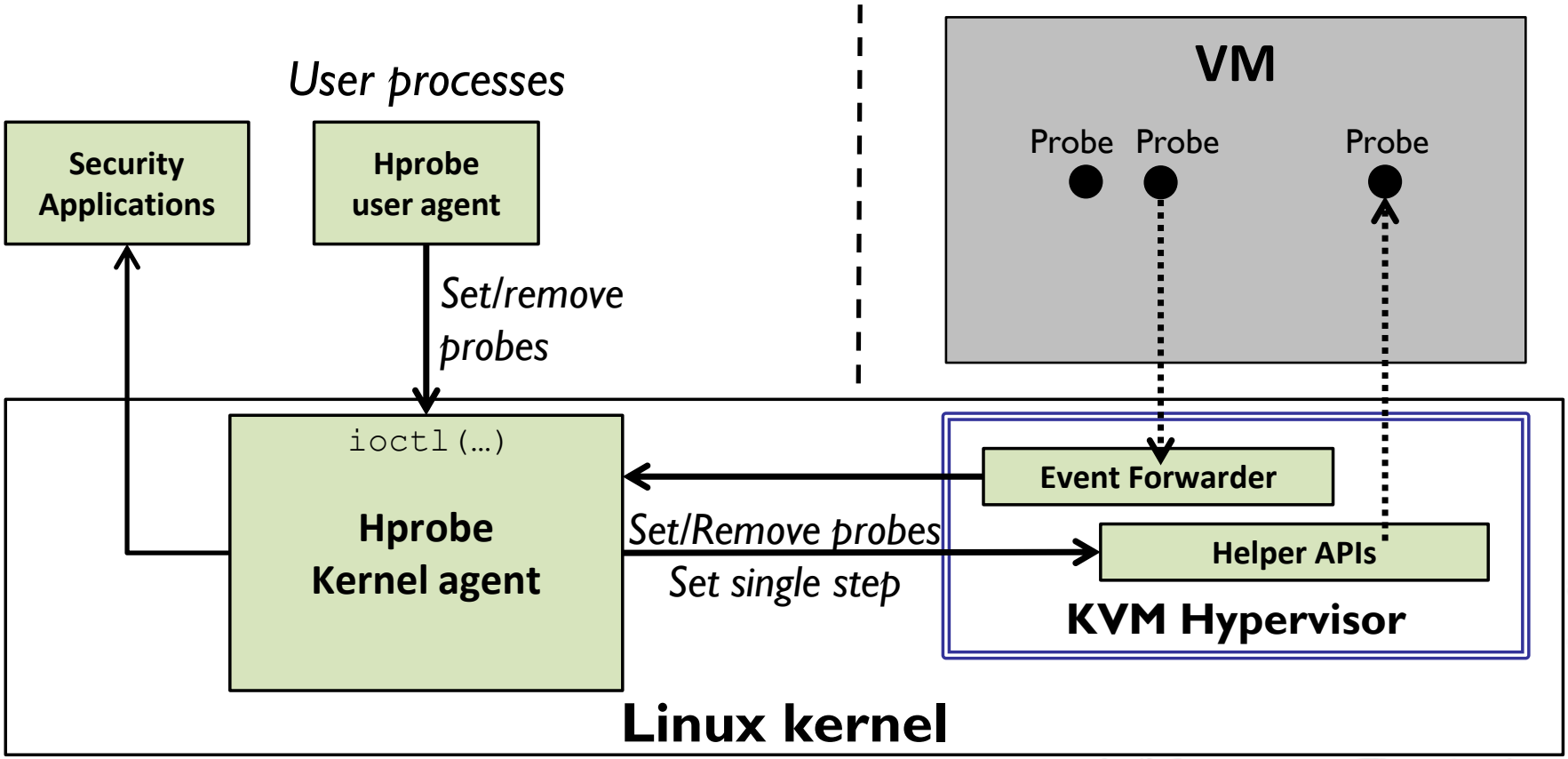
Hprobes

- Active monitoring
- Transparent - OS agnostic
- Dynamic
- Protected

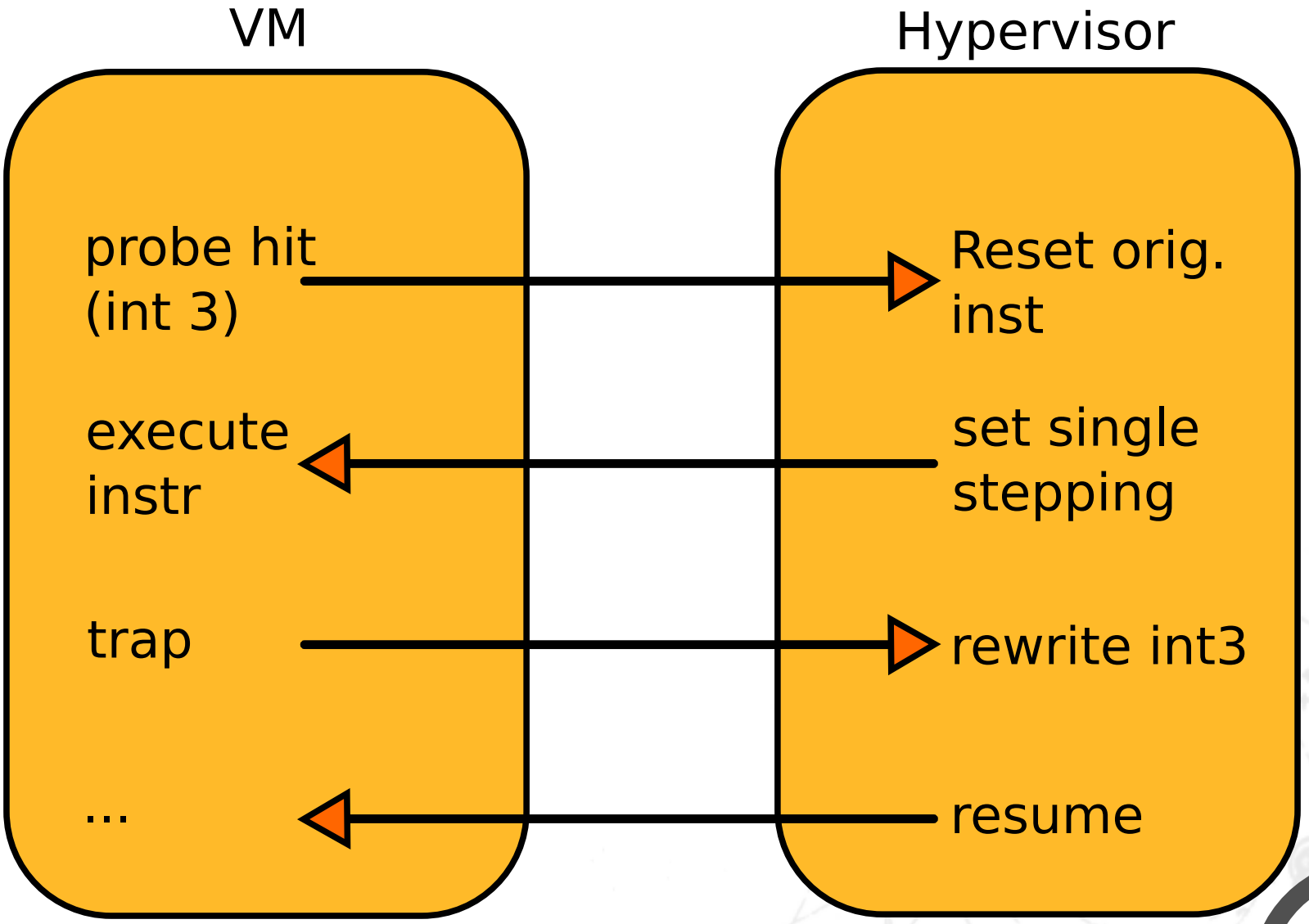
Hprobes

- Active monitoring
- Transparent - OS agnostic
- Dynamic
- Protected
- Simple and Fast

Architecture

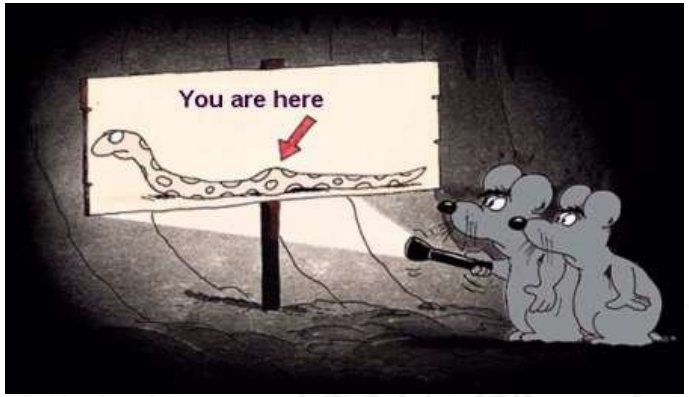


Hitting a probe



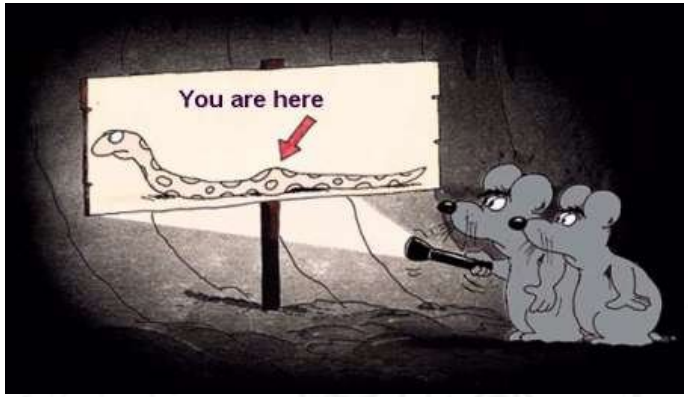
Where we are now

✓ Kernel-space probes



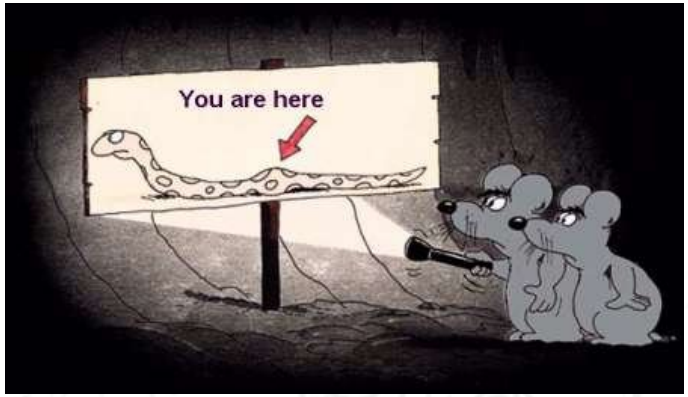
Where we are now

- ✓ Kernel-space probes
- ✓ Kernel patch



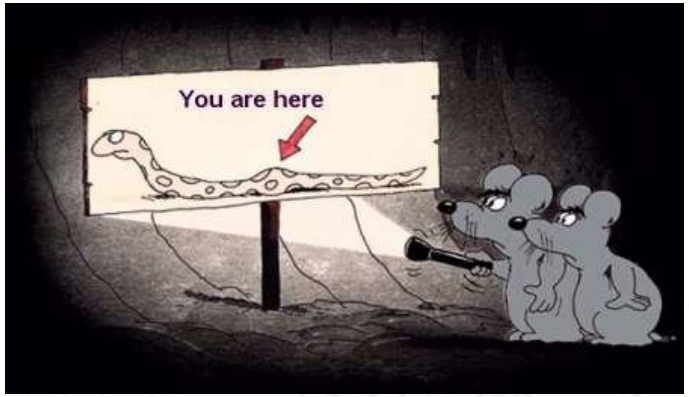
Where we are now

- ✓ Kernel-space probes
- ✓ Kernel patch
- ✓ CLI Useragent



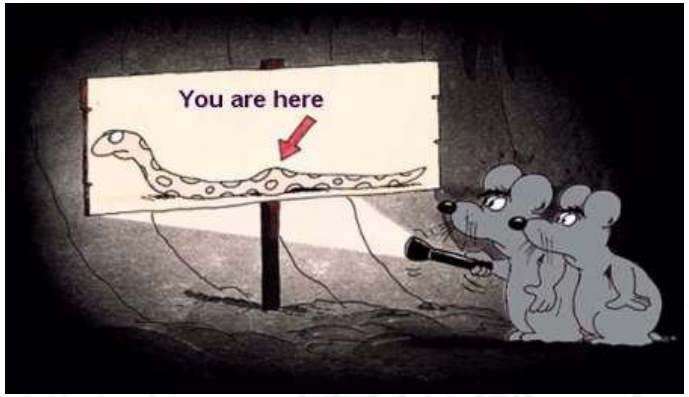
Where we are now

- ✓ Kernel-space probes
- ✓ Kernel patch
- ✓ CLI Useragent
  - ✓ integration with libvmi



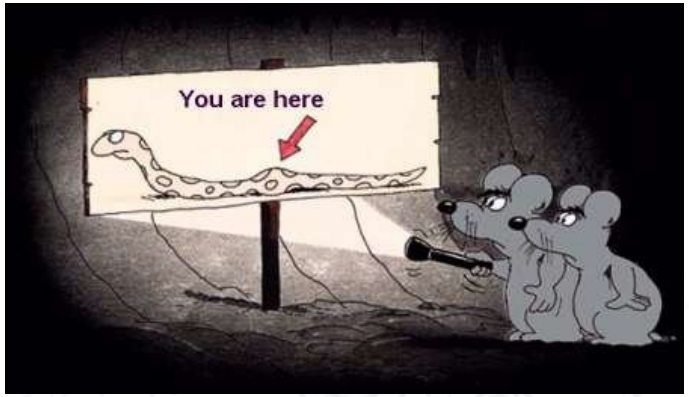
Where we are now

- ✓ Kernel-space probes
- ✓ Kernel patch
- ✓ CLI Useragent
  - ✓ integration with libvmi
- → User-space probes



Where we are now

- ✓ Kernel-space probes
- ✓ Kernel patch
- ✓ CLI Useragent
  - ✓ integration with libvmi
- → User-space probes
  - ○ Then actual applications



Potential Uses

What can we do with this?

Potential Uses

What can we do with this?

- Arbitrarily specify events
- More app-level monitoring

Potential Uses

Heartbeat/watchdog

- App-level

Potential Uses

Heartbeat/watchdog

- App-level
- Critical code sections

Potential Uses

Heartbeat/watchdog

- App-level
- Critical code sections
- Act?

Potential Uses

Avoid TOCTTOU race

Potential Uses

Avoid TOCTTOU race

```
int fd;
if (access(argv[1], R_OK) != 0)
exit(1);
fd = open(argv[1], O_RDONLY);
```

Potential Uses

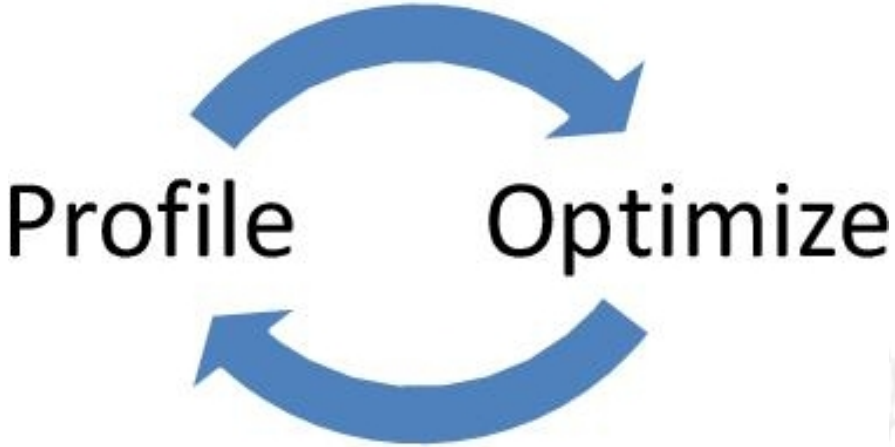
Avoid TOCTTOU race

```
int fd;
if (access(argv[1], R_OK) != 0)
exit(1);
fd = open(argv[1], O_RDONLY);
```

- Can check exactly at time of use

Potential Uses

Guest profiling (think Systemtap)



Potential Uses

Process-specific tests

Potential Uses

Process-specific tests

- Custom to application

Potential Uses

Process-specific tests

- Custom to application
- User supplied?

Potential Uses

Process-specific tests

- Custom to application
- User supplied?
- Cloud provider perspective?

Potential Uses

Execution artifacts (signature)

Potential Uses

Execution artifacts (signature)

- Place probes in applications

Potential Uses

Execution artifacts (signature)

- Place probes in applications
- Characterize by execution pattern?

Potential Uses

Execution artifacts (signature)

- Place probes in applications
- Characterize by execution pattern?
- Invariants?

Performance

Hook/probe Latency

Hook/probe Latency

- Lares: 27 μs/hook (2.3 GHz CPU)
- hprobes: 3.2 μs/probe (3.0GHz CPU)
- SIM: 0.4μs/event (2.4 GHz CPU)

Hook/probe Latency

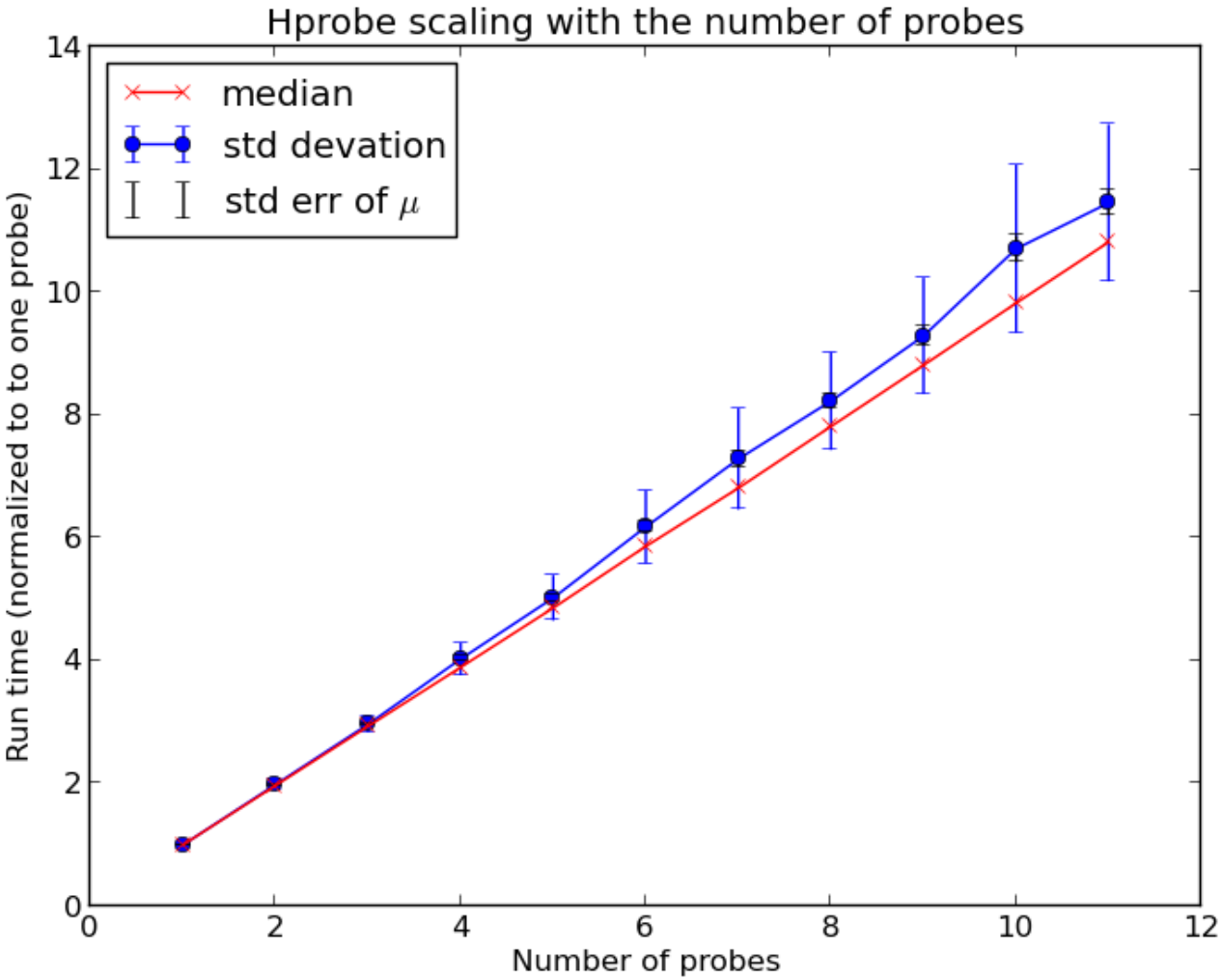
- Lares: 27 μs/hook (2.3 GHz CPU)
- hprobes: 3.2 μs/probe (3.0GHz CPU)
- SIM: 0.4μs/event (2.4 GHz CPU)

No tuning/perf modifications

Some quick benchmarks

Test Name	Run time (s) w/o probes*	Run time (s) w/ probes*	Percent Difference	Probe locations
find /	88.69 (0.27)	92.10 (1.98)	3.82%	sys_open
SPECjvm2008	360.5 (7.50)	492.43 (19.4)	36.6%	sys_open sys_read sys_write sys_close
kernel make	245.15 (16.7)	250.06 (14.5)	2.00%	sys_open
kernel make	245.15 (16.7)	245.41 (16.3)	0.11%	sys_write
kernel make	245.15 (16.7)	245.13 (16.7)	0.01%	sys_read
kernel make	245.15 (16.7)	246.38 (16.7)	0.50%	sys_close

A Scaling Picture



Summary

- Hooks
- Lares, SIM, etc...
- How hprobes work
- Current status
- Potential uses
- Early performance data

Summary

- Hooks
- Lares, SIM, etc...
- How hprobes work
- Current status
- Potential uses
- Early performance data

Comments/questions/ideas?

LaTeX beamer theme courtesy Flip Tanedo, Cornell

Images are rights of their respective owners