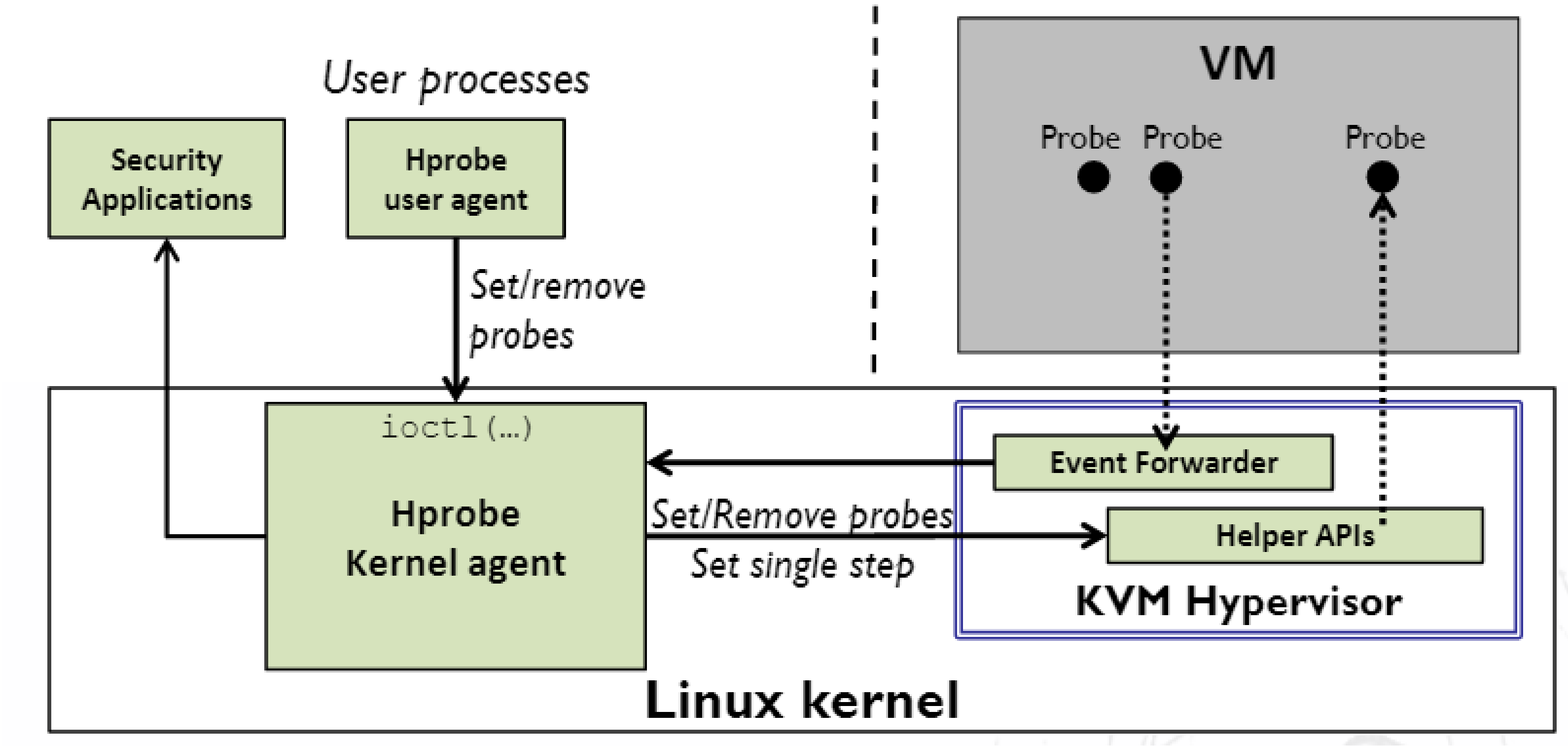
Formalizing Hardware-Assisted Virtualization Behavior to Verify VM Monitoring Frameworks

Lavin Devnani

ACC Seminar – April 5 2017

Dynamic VM Monitoring - Hprobes



Z. J. Estrada, C. Pham, F. Deng, L. Yan, Z. Kalbarczyk and R. K. Iyer, "Dynamic VM Dependability Monitoring Using Hypervisor Probes," *2015 11th European Dependable Computing Conference (EDCC)*, Paris, 2015, pp. 61-72.

VM Monitoring Framework Integration

Integrate Framework

- Hypervisor Extension
- Kernel Module
- User APIs/Interfaces

Develop Custom Monitors & Auditors

- Develop custom monitors according to framework specifications

Verify correctness, security properties of specified monitors

- Specify monitor behavior in formal terms
- Define trigger events, alarm conditions, probe locations
- Run verification tool

Deploy specified monitors

- Make monitors available to users

Entity Description

- We build our model using "entities"

$$E = (\mathcal{F}, \mathcal{P})$$

- Where
  - $\mathcal{F}$ is an ordered list of system events that represents the execution flow of that entity
  - $\mathcal{P}$ is a set of properties/variables of that entity
- Example

  Hardware

  $= (\ movCR3 \rightarrow LIDT \rightarrow LGDT, \{\ General\ Purpose\ Regs,\ \ Control\ Regs,\ \ Segment\ Regs\ \}\ )$

Entity Description - $E = (\mathcal{F}, \mathcal{P})$

- Given a set of entities, a system event can
  - Modify value of some property $p \in \mathcal{P}$
  - Generate new system events in current execution flow, $\mathcal{F}$
  - Generate new system events in other entities
- Behavior of each event described using rewrite rules in Maude
- Example

```
< UserEntity | Flow : sys_read(args) , Properties: … >
< KernelEntity | Flow : nil, Properties: … >
   =>
< UserEntity | Flow : nil, Properties: … >
< KernelEntity | Flow : system_call call_jump_table read_syscall,
                 Properties: … >
```

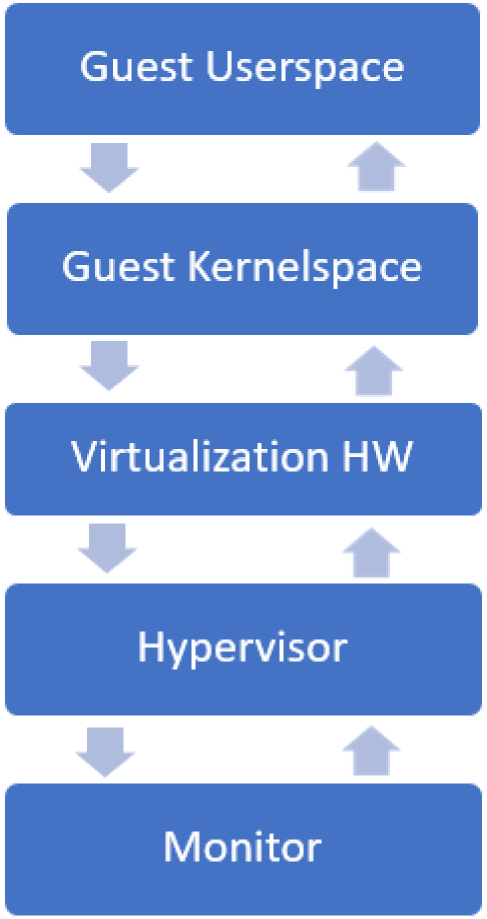
Model Description

- We formalize system behavior as 5 entities

$$S = (\mathcal{U}, \mathcal{K}, \mathcal{V}, \mathcal{H}, \mathcal{M})$$

where

- $\mathcal{U}$ - guest user space behavior
- $\mathcal{K}$ - guest kernel behavior
- $\mathcal{V}$ - hardware virtualization variables
- $\mathcal{H}$ - hypervisor behavior
- $\mathcal{M}$ - monitor



Monitor Verification

- We define a monitor as conditional rewrite rules with
  - Inputs – Properties from other entities
  - Outputs – Alert flag
- For a given monitor & probes, we run state space searches to identify
  - Execution flows that lead to monitor alerts
  - Missing probe locations, if any
  - Logic bugs in monitor specifications

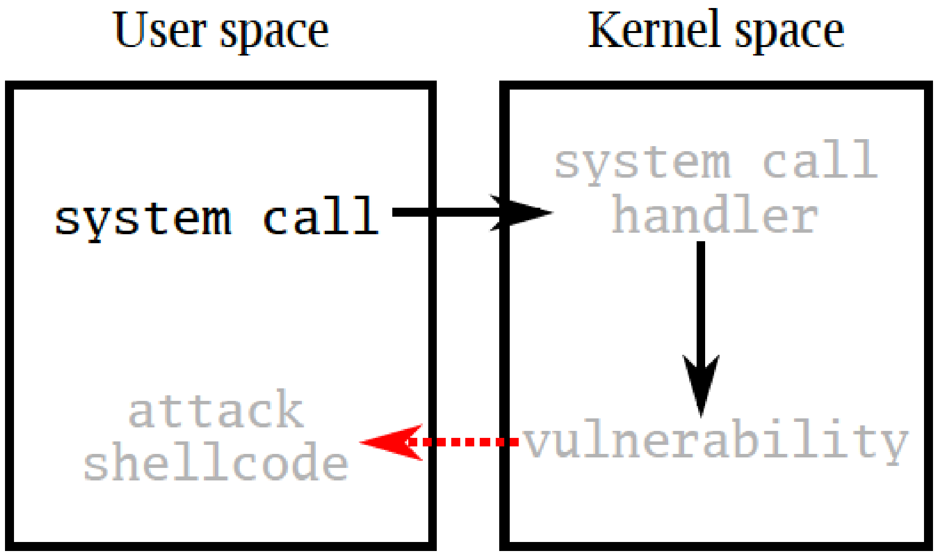
Example Execution – Return to User Attack

```
init_monitor:
    attack_detected = false
    return

probe_hit(context):
    if (context.CPL == KERNEL &&
        userPageExecuting)
       attack_detected = true
       raise_alert()
    return
```

- Return2User monitor places probe in every kernel entry and exit point
- Each probe invocation identifies whether user or kernel page is running
- Alert is raised when a user page executes with kernel permissions

Example Execution – Return to User Attack



System Call handler exploited via Ret2User attack

CVEs 2008-0600, 2009-2692, 2009-3547, 2010-4258

```
< USER | ExecFlow: UserCodeBlock
                   SyscallRead
                   AttackShellCode,
         Properties: nil >

< KERNEL | ExecFlow: SystemCallHandler
                     Vulnerability,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = USER, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: User>
```

```
< USER | ExecFlow: UserCodeBlock
                   SyscallRead
                   AttackShellCode,
         Properties: nil >

< KERNEL | ExecFlow: (PROBE SystemCallHandler)
                      Vulnerability,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = USER, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: User>
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: UserCodeBlock
                   SyscallRead
                   AttackShellCode,
         Properties: nil >

< KERNEL | ExecFlow: (PROBE SystemCallHandler)
                      Vulnerability,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = USER, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: User>
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: UserCodeBlock
                   SyscallRead
                   AttackShellCode,
         Properties: nil >

< KERNEL | ExecFlow: (PROBE SystemCallHandler)
                      Vulnerability,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = USER, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: User>
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: SyscallRead
                   AttackShellCode,
         Properties: nil >


< KERNEL | ExecFlow: (PROBE SystemCallHandler)
                      Vulnerability,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = USER, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: User>
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
      (MON.AttackDetected = false)
```

```
< USER | ExecFlow: SyscallRead
                   AttackShellCode,
         Properties: nil >


< KERNEL | ExecFlow: (PROBE SystemCallHandler)
                      Vulnerability,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = USER, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: User>
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

< **USER** | ExecFlow: **AttackShellCode**,
Properties: nil >

< **KERNEL** | ExecFlow: **(PROBE SystemCallHandler)**
Vulnerability,
Properties: … >

< **HW** | ExecFlow: nil, Properties: **CPL = KERNEL**, … >

< **HYP** | ExecFlow: nil, Properties: … >

< **MON** | Return2UserMonitor AttackDetected: false, Active: User>

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: AttackShellCode,
         Properties: nil >



< KERNEL | ExecFlow: (PROBE SystemCallHandler)
                      Vulnerability,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = KERNEL, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: User>
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

< **USER** | ExecFlow: **AttackShellCode**,
Properties: nil >

< **KERNEL** | ExecFlow: **SystemCallHandler Vulnerability**,
Properties: … >

< **HW** | ExecFlow: **Int3**, Properties: **CPL = KERNEL**, … >

< **HYP** | ExecFlow: nil, Properties: … >

< **MON** | Return2UserMonitor AttackDetected: false, Active: User >

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: AttackShellCode,
         Properties: nil >

< KERNEL | ExecFlow: SystemCallHandler
                      Vulnerability,
           Properties: … >

< HW | ExecFlow: Int3, Properties: CPL = KERNEL, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: User >
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
      (MON.AttackDetected = false)
```

< **USER** | ExecFlow: **AttackShellCode**,
Properties: nil >

< **KERNEL** | ExecFlow: **SystemCallHandler Vulnerability**,
Properties: … >

< **HW** | ExecFlow: nil, Properties: **CPL = KERNEL**, … >

< **HYP** | ExecFlow: **VMExit**, Properties: … >

< **MON** | Return2UserMonitor AttackDetected: false, Active: User >

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: AttackShellCode,
         Properties: nil >



< KERNEL | ExecFlow: SystemCallHandler
                      Vulnerability,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = KERNEL, … >

< HYP | ExecFlow: VMExit, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: User >
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: AttackShellCode,
         Properties: nil >



< KERNEL | ExecFlow: SystemCallHandler
                      Vulnerability,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = KERNEL, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: Kernel >
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
      (MON.AttackDetected = false)
```

```
< USER | ExecFlow: AttackShellCode,
         Properties: nil >


< KERNEL | ExecFlow: SystemCallHandler
                      Vulnerability,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = KERNEL, … >

< HYP | ExecFlow: VMEntry, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: Kernel >
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: AttackShellCode,
         Properties: nil >



< KERNEL | ExecFlow: SystemCallHandler
                      Vulnerability,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = KERNEL, … >

< HYP | ExecFlow: VMEntry, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: Kernel >
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: AttackShellCode,
         Properties: nil >



< KERNEL | ExecFlow: SystemCallHandler
                      Vulnerability,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = KERNEL, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: Kernel >
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: AttackShellCode,
         Properties: nil >

< KERNEL | ExecFlow: SystemCallHandler
                     Vulnerability,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = KERNEL, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: Kernel >
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: AttackShellCode,
         Properties: nil >


< KERNEL | ExecFlow: Vulnerability,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = KERNEL, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: Kernel >
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: AttackShellCode,
         Properties: nil >

< KERNEL | ExecFlow: Vulnerability,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = KERNEL, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: Kernel >
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: AttackShellCode,
         Properties: nil >



< KERNEL | ExecFlow: nil,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = KERNEL, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: Kernel >
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
      (MON.AttackDetected = false)
```

```
< USER | ExecFlow: AttackShellCode,
         Properties: nil >

< KERNEL | ExecFlow: nil,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = KERNEL, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: Kernel>
```

```
< USER | ExecFlow: UserCodeBlock
                   SyscallRead
                   (PROBE AttackShellCode) ,
         Properties: nil >

< KERNEL | ExecFlow: (PROBE SystemCallHandler)
                      Vulnerability,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = USER, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: User>
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: UserCodeBlock
                   SyscallRead
                   (PROBE AttackShellCode),
         Properties: nil >

< KERNEL | ExecFlow: (PROBE SystemCallHandler)
                      Vulnerability,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = USER, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: User>
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: UserCodeBlock
                   SyscallRead
                   (PROBE AttackShellCode),
         Properties: nil >

< KERNEL | ExecFlow: (PROBE SystemCallHandler)
                      Vulnerability,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = USER, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: User>
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
      (MON.AttackDetected = false)
```

```
< USER | ExecFlow: SyscallRead
                   (PROBE AttackShellCode),
         Properties: nil >


< KERNEL | ExecFlow: (PROBE SystemCallHandler)
                      Vulnerability,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = USER, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: User>
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: SyscallRead
                   (PROBE AttackShellCode),
         Properties: nil >


< KERNEL | ExecFlow: (PROBE SystemCallHandler)
                      Vulnerability,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = USER, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: User>
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
      (MON.AttackDetected = false)
```

< **USER** | ExecFlow: **(PROBE AttackShellCode)**,
Properties: nil >

< **KERNEL** | ExecFlow: **(PROBE SystemCallHandler)**
Vulnerability,
Properties: … >

< **HW** | ExecFlow: nil, Properties: **CPL = KERNEL**, … >

< **HYP** | ExecFlow: nil, Properties: … >

< **MON** | Return2UserMonitor AttackDetected: false, Active: User>

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: (PROBE AttackShellCode),
         Properties: nil >
```

```
< KERNEL | ExecFlow: (PROBE SystemCallHandler)
                      Vulnerability,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = KERNEL, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: User>
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

< **USER** | ExecFlow: **(PROBE AttackShellCode)**,
Properties: nil >

< **KERNEL** | ExecFlow: **SystemCallHandler Vulnerability**,
Properties: … >

< **HW** | ExecFlow: **Int3**, Properties: **CPL = KERNEL**, … >

< **HYP** | ExecFlow: nil, Properties: … >

< **MON** | Return2UserMonitor AttackDetected: false, Active: User >

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: (PROBE AttackShellCode),
         Properties: nil >



< KERNEL | ExecFlow: SystemCallHandler
                      Vulnerability,
           Properties: … >

< HW | ExecFlow: Int3, Properties: CPL = KERNEL, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: User >
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: (PROBE AttackShellCode),
         Properties: nil >



< KERNEL | ExecFlow: SystemCallHandler
                      Vulnerability,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = KERNEL, … >

< HYP | ExecFlow: VMExit, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: User >
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
      (MON.AttackDetected = false)
```

```
< USER | ExecFlow: (PROBE AttackShellCode),
         Properties: nil >


< KERNEL | ExecFlow: SystemCallHandler
                      Vulnerability,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = KERNEL, … >

< HYP | ExecFlow: VMExit, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: User>
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

< **USER** | ExecFlow: **(PROBE AttackShellCode)**,
Properties: nil >

< **KERNEL** | ExecFlow: **SystemCallHandler**
Vulnerability,
Properties: … >

< **HW** | ExecFlow: nil, Properties: **CPL = KERNEL**, … >

< **HYP** | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: Kernel >

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

< **USER** | ExecFlow: **(PROBE AttackShellCode)**,
Properties: nil >

< **KERNEL** | ExecFlow: **SystemCallHandler**
Vulnerability,
Properties: … >

< **HW** | ExecFlow: nil, Properties: **CPL = KERNEL**, … >

< **HYP** | ExecFlow: **VMEntry**, Properties: … >

< **MON** | Return2UserMonitor AttackDetected: false, Active: Kernel >

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: (PROBE AttackShellCode),
         Properties: nil >

< KERNEL | ExecFlow: SystemCallHandler
                      Vulnerability,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = KERNEL, … >

< HYP | ExecFlow: VMEntry, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: Kernel >
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: (PROBE AttackShellCode),
         Properties: nil >

< KERNEL | ExecFlow: SystemCallHandler
                      Vulnerability,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = KERNEL, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: Kernel >
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: (PROBE AttackShellCode),
         Properties: nil >



< KERNEL | ExecFlow: SystemCallHandler
                     Vulnerability,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = KERNEL, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: Kernel >
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
      (MON.AttackDetected = false)
```

< **USER** | ExecFlow: **(PROBE AttackShellCode)**,
Properties: nil >

< **KERNEL** | ExecFlow: **Vulnerability**,
Properties: … >

< **HW** | ExecFlow: nil, Properties: **CPL = KERNEL**, … >

< **HYP** | ExecFlow: nil, Properties: … >

< **MON** | Return2UserMonitor AttackDetected: false, Active: Kernel >

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: (PROBE AttackShellCode),
         Properties: nil >


< KERNEL | ExecFlow: Vulnerability,
           Properties: … >


< HW | ExecFlow: nil, Properties: CPL = KERNEL, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: Kernel >
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: (PROBE AttackShellCode),
         Properties: nil >
```

```
< KERNEL | ExecFlow: nil,
           Properties: … >
```

```
< HW | ExecFlow: nil, Properties: CPL = KERNEL, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: Kernel >
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: AttackShellCode,
         Properties: nil >



< KERNEL | ExecFlow: nil,
           Properties: … >

< HW | ExecFlow: Int3, Properties: CPL = KERNEL, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: Kernel>
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: AttackShellCode,
         Properties: nil >



< KERNEL | ExecFlow: nil,
           Properties: … >

< HW | ExecFlow: Int3, Properties: CPL = KERNEL, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: Kernel >
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

< **USER** | ExecFlow: **AttackShellCode**,
Properties: nil >

< **KERNEL** | ExecFlow: nil,
Properties: … >

< **HW** | ExecFlow: nil, Properties: **CPL = KERNEL**, … >

< **HYP** | ExecFlow: **VMExit**, Properties: … >

< **MON** | Return2UserMonitor AttackDetected: false, Active: Kernel >

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: AttackShellCode,
         Properties: nil >



< KERNEL | ExecFlow: nil,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = KERNEL, … >

< HYP | ExecFlow: VMExit, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: Kernel >
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: AttackShellCode,
         Properties: nil >



< KERNEL | ExecFlow: nil,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = KERNEL, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: User >
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: AttackShellCode,
         Properties: nil >

< KERNEL | ExecFlow: nil,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = KERNEL, … >

< HYP | ExecFlow: VMEntry, Properties: … >

< MON | Return2UserMonitor AttackDetected: false, Active: User >
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

< **USER** | ExecFlow: **AttackShellCode**,
Properties: nil >

< **KERNEL** | ExecFlow: nil,
Properties: … >

< **HW** | ExecFlow: nil, Properties: **CPL = KERNEL**, … >

< **HYP** | ExecFlow: **VMEntry**, Properties: … >

< **MON** | Return2UserMonitor AttackDetected: false, Active: User >

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: AttackShellCode,
         Properties: nil >


< KERNEL | ExecFlow: nil,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = KERNEL, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: true, Active: User >
```

```
Maude> Search (USER.ExecFlow = AttackShellCode) /\ (HW.CPL = KERNEL) /\
       (MON.AttackDetected = false)
```

```
< USER | ExecFlow: nil,
         Properties: nil >
```

```
< KERNEL | ExecFlow: nil,
           Properties: … >

< HW | ExecFlow: nil, Properties: CPL = KERNEL, … >

< HYP | ExecFlow: nil, Properties: … >

< MON | Return2UserMonitor AttackDetected: true, Active: User >
```

Future Work

- Expand model specification to cover additional kernel events
- Verify behavior of additional monitors
- Support verification of other frameworks (*HyperTap*)
- Workshop Paper

Questions