Truman: Constructing Device Behavior Models from OS Drivers to Fuzz Virtual Devices

Zheyu Ma[1,3,5], Qiang Liu[3], Zheming Li[1,5], Tingting Yin[2], Wende Tan[4],

Chao Zhang[1,2,5,*], Mathias Payer[3]

1. Institute for Network Sciences and Cyberspace (INSC), Tsinghua University 2.Zhongguancun Laboratory

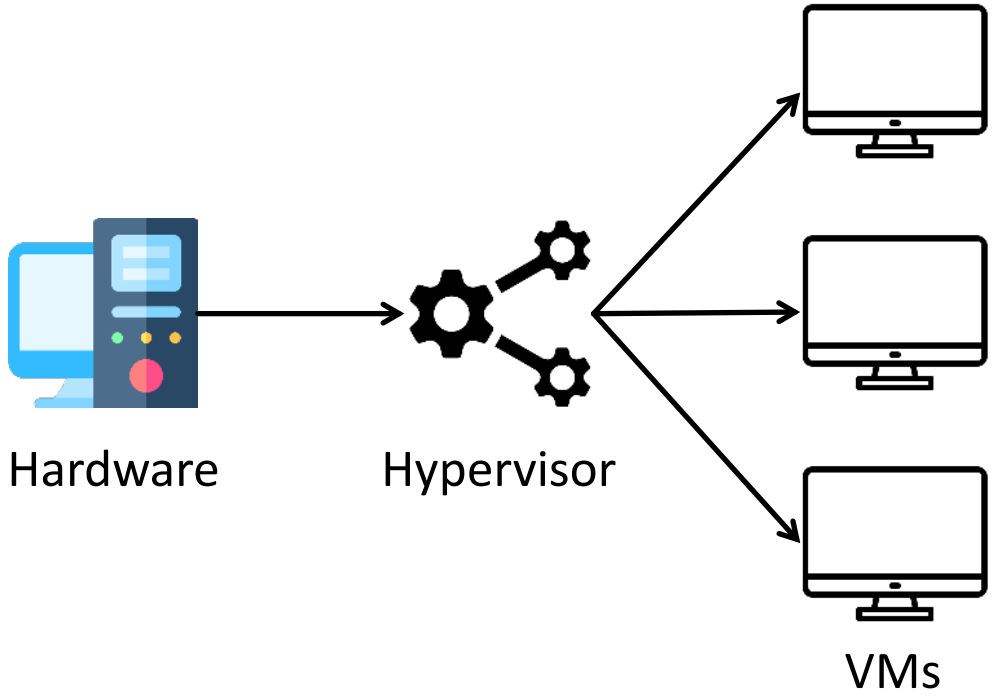
3. EPFL 4. Department of Computer Science and Technology, Tsinghua University

5. JCSS, Tsinghua University (INSC) - Science City (Guangzhou) Digital Technology Group Co., Ltd.

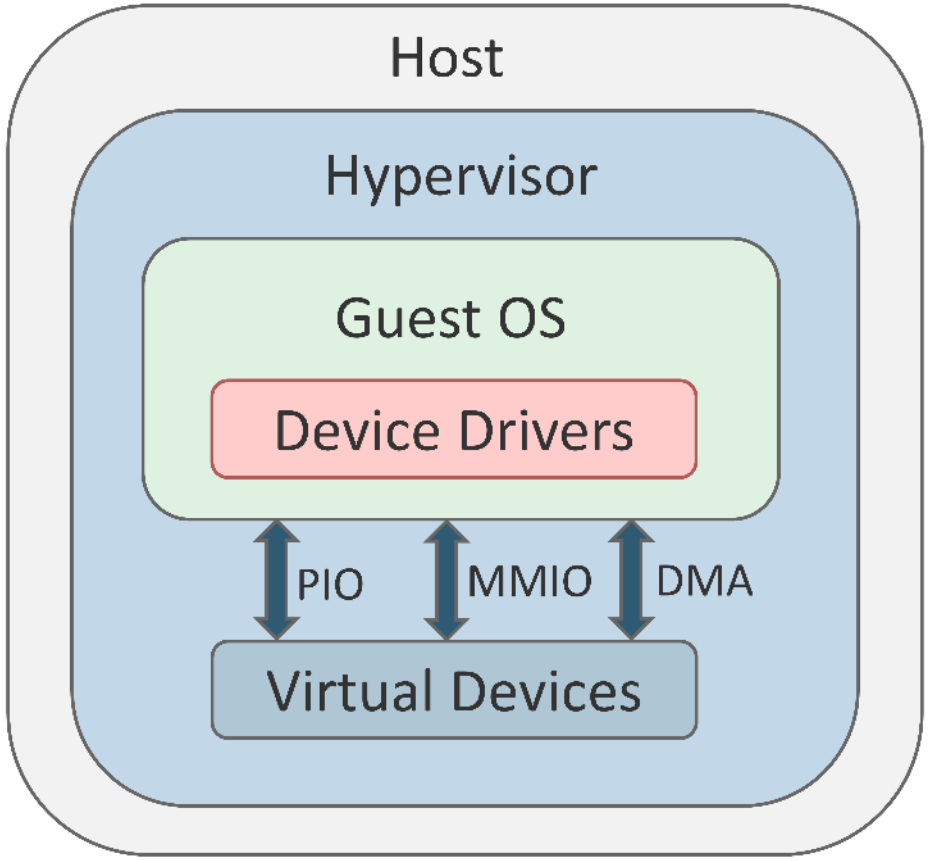
* Corresponding Author

Motivation: Hypervisor Security is Essential

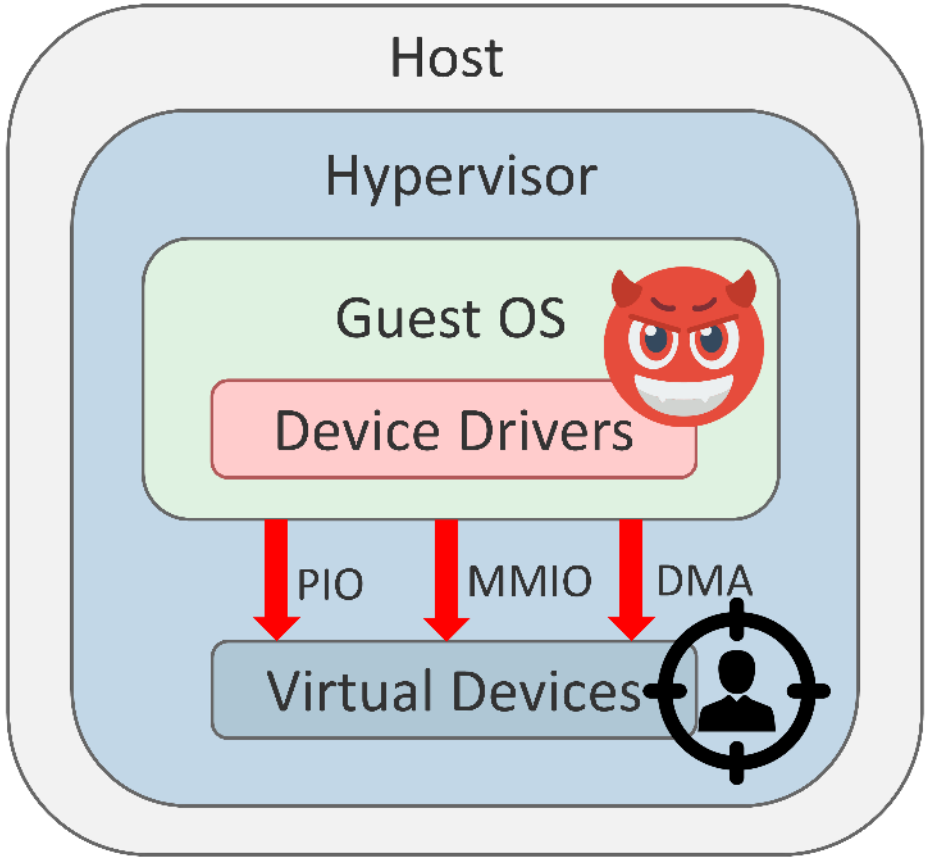
The **Hypervisor** manages VMs and is a key component in cloud computing.



Hypervisor Threat Model



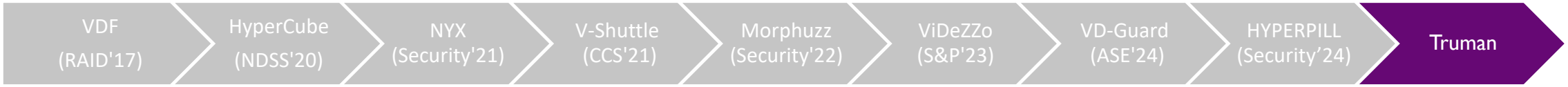
Hypervisor Threat Model



Past Research on Virtual Devices

- **Interface:** VDF, HyperCube
- **Fuzzing Engine:** HyperCube, NYX, HYPERPILL
- **Input Grammar:** V-Shuttle, Morphuzz, HYPERPILL
- **High-Quality Testcases:** ViDeZZo, VD-Guard

Past Research on Virtual Devices



- **Interface:** VDF, HyperCube
- **Fuzzing Engine:** HyperCube, NYX, HYPERPILL
- **Input Grammar:** V-Shuttle, Morphuzz, HYPERPILL
- **High-Quality Testcases:** ViDeZZo, VD-Guard, Truman

Challenge 1: Devices Require Complex Interaction

The **order** of virtual device messages
e.g., Setup -> Configuration -> Operational -> Cleanup

Efficient fuzzing needs **automation** to extract **inter- and intra-message dependencies** without manual help (NYX), source of virtual devices (ViDeZZo), or random exploration (Morphuzz).

Challenge 1: Devices Require Complex Interaction

1. Constraints on a single field. e.g., the components of a register
2. Relationships between fields. e.g., nested DMA buffers

The **order** of virtual device messages
e.g., Setup -> Configuration -> Operational -> Cleanup

Efficient fuzzing needs **automation** to extract **inter- and intra-message dependencies** without manual help (NYX), source of virtual devices (ViDeZZo), or random exploration (Morphuzz).

Challenge 2: Some Devices are Hidden Behind a Bus

The **set** of inter- and intra- message dependencies that are valid in a specific state

Exploring **bus-hidden** devices requires the device's **state dependency**

to guide the exploration.

Challenge 2: Some Devices are Hidden Behind a Bus

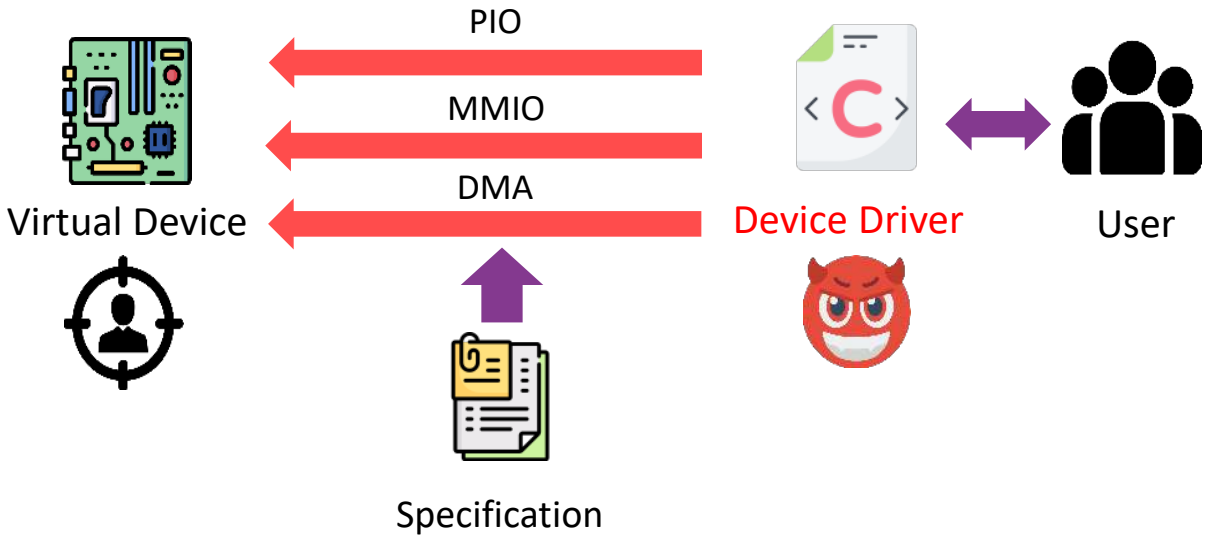
The **set** of inter- and intra- message dependencies that are valid in a specific state

The interfaces of devices are hidden by the bus. e.g., virtio

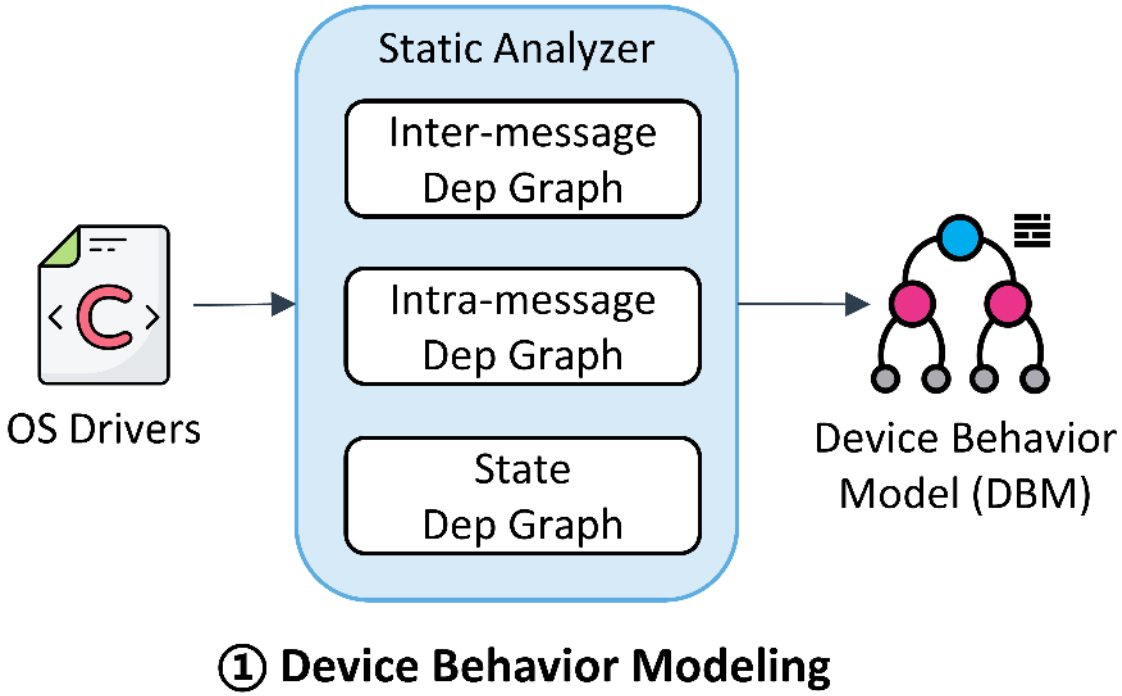
Exploring **bus-hidden** devices requires the device's **state dependency** to guide the exploration.

Insight: Extract Knowledge from Device Driver

Each (closed-source) virtual device and its corresponding open-source driver follow their shared device specification

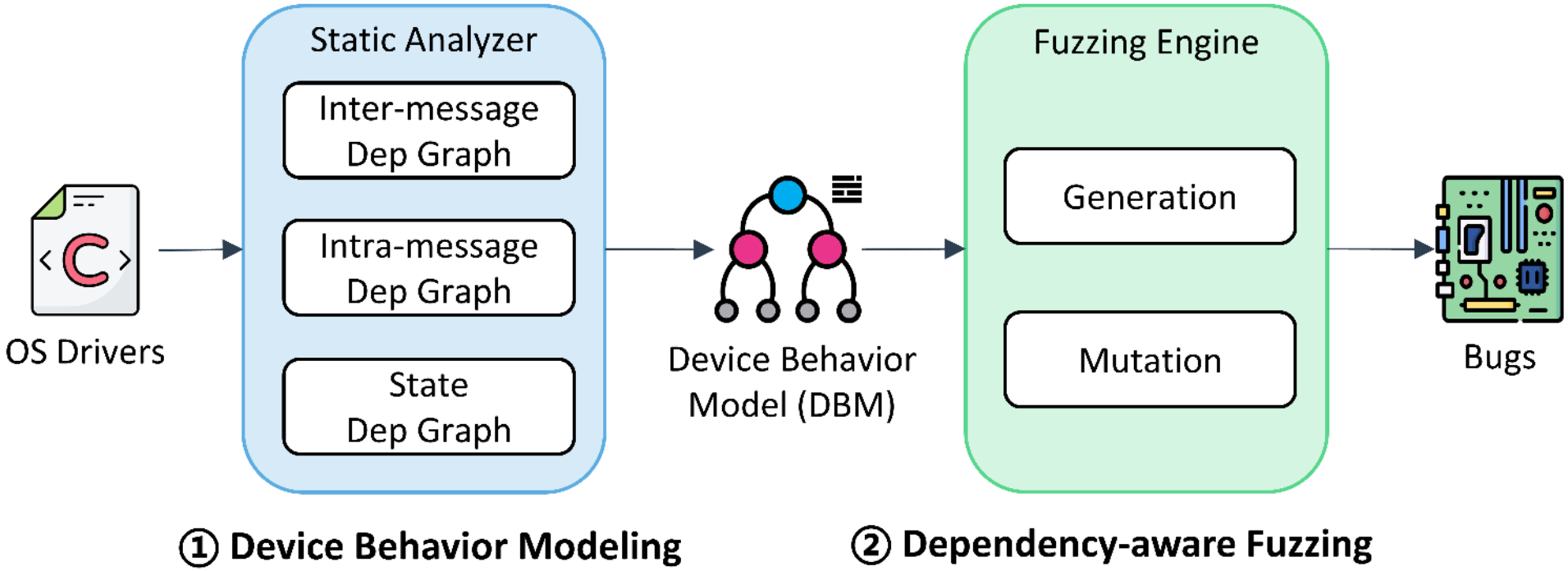


Truman Design: Two-step Framework



① Device Behavior Modeling

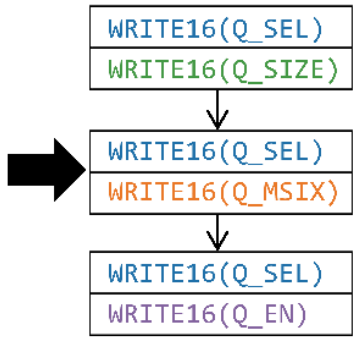
Truman Design: Two-step Framework



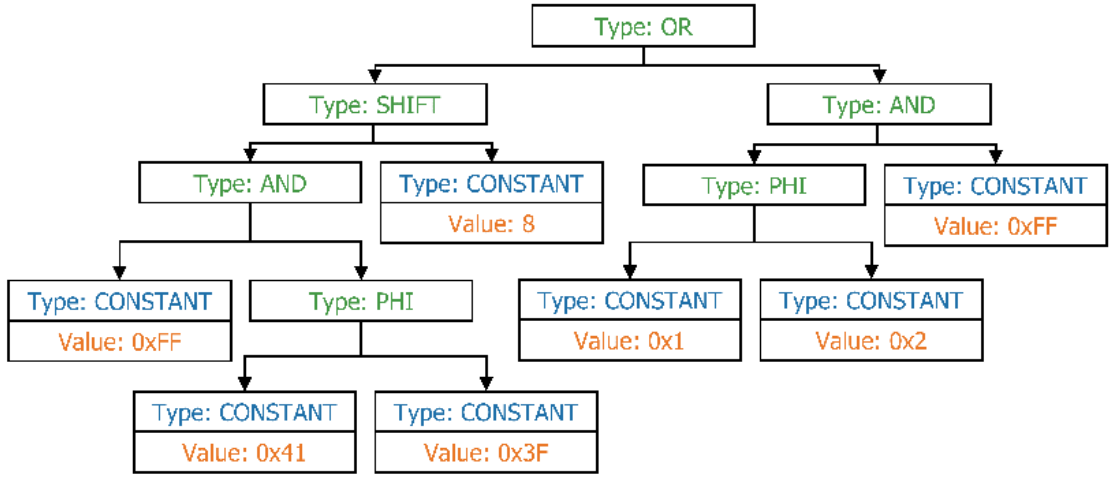
Inter-/intra-message Dependency

```
vp_iowrite16(index, &mdev->common->queue_select);
vp_iowrite16(size, &mdev->common->queue_size);

vp_iowrite16(index, &cfg->queue_select);
vp_iowrite16(vector, &cfg->queue_msix_vector);

vp_iowrite16(index, &mdev->common->queue_select);
vp_iowrite16(enable, &mdev->common->queue_enable);
```

Driver of Virtio Bus



InterDep Graph



Inter-message dependency:

CG/CFG Traversal

Intra-message dependency:

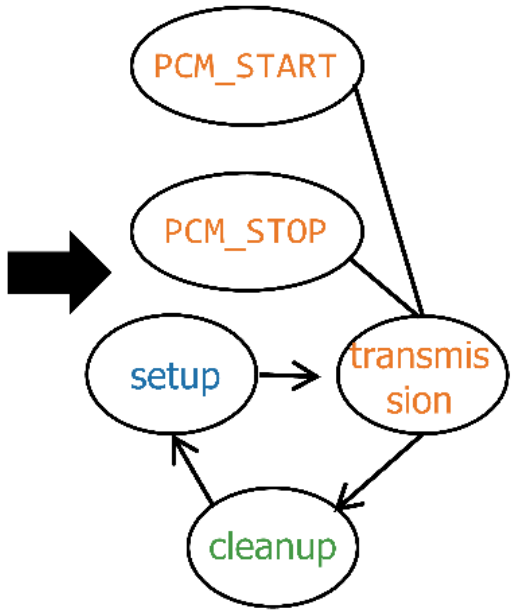
Backward dataflow analysis

State Dependency

```
struct virtio_driver = {
  .name       = "virtio",
  .probe      = virtio_probe,
  .remove     = virtio_remove,
}
```

```
struct message_header *hdr = message->header;
/* command could be
   VIRTIO_SND_R_PCM_START or VIRTIO_SND_R_PCM_STOP */
hdr->hdr.code = cpu_to_le32(command);
hdr->stream_id = cpu_to_le32(vss->sid);
```

Virtio Driver



StateDep Graph

State dependency:

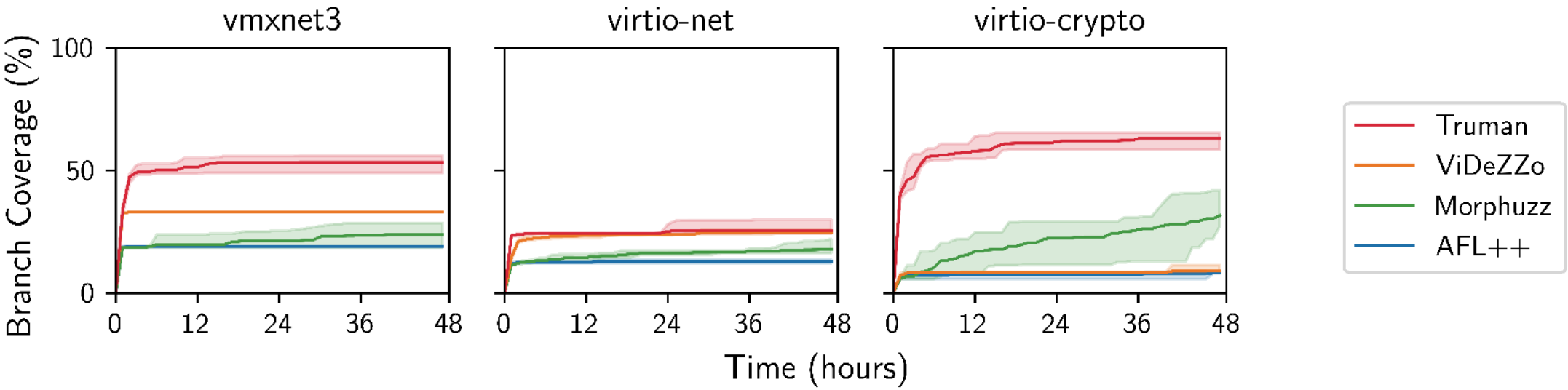
Analyze the bus driver and the device driver

Dependency-aware Fuzzing

- Generation: Message level, BB level, Function level, State level
- Mutation: Message level, Sequence level, State level
- Execution: Develop a platform-agnostic dependency-aware fuzzing engine to (de)serialize the virtual device messages

Evaluation: Code Coverage

Truman outperforms **19/29** devices over AFL++, Morphuzz, and ViDeZZo.



Evaluation: Vulnerability Discovery

Known bug discovery

Devices	Morphuzz	ViDeZZo	Truman
intel-hda	1	1	1
am53c974	0	1	1
ide-hd	1	0	2
nvme	0	0	2
virtio-net	1	1	1
virtio-gpu	0	0	1
sm501	1	1	2
Total	4	4	10

Evaluation: Vulnerability Discovery

Known bug discovery

Devices	Morphuzz	ViDeZZo	Truman
intel-hda	1	1	1
am53c974	0	1	1
ide-hd	1	0	2
nvme	0	0	2
virtio-net	1	1	1
virtio-gpu	0	0	1
sm501	1	1	2
Total	4	4	10

New bug discovery

- QEMU, VirtualBox, VMware Workstation, Parallels
- 54 bugs found
- 6 CVEs

Truman: Constructing Device Behavior Models from OS Drivers to Fuzz Virtual Devices

- Truman proposes an **automatic** approach to extract inter-/intra-message dependencies and state dependencies from open-source **OS drivers**
- Truman found **54 bugs** in 4 major hypervisors, 6 CVEs
- Code at https://github.com/vul337/Truman