Hyper-Cube
High-Dimensional Hypervisor Fuzzing
Sergej Schumilo, Cornelius Aschermann, Ali Abbasi, Simon Wörner and Thorsten Holz
Chair for Systems Security
Ruhr-Universität Bochum
Motivation

Hypervisor

Malicious Guest (Privileged; Running in Ring-0)

Local VM DoS (Crash or Deadlock)

Virtual Machine DoS (Crash or Deadlock)

Virtual Machine Escape (Other Guest)

Host DoS (Kernel Panic or Deadlock)

Virtual Machine Escape (Host)
Motivation

Hypervisor

VM 1

VM 2

Malicious Guest (Privileged; Running in Ring-0)

Local VM DoS (Crash or Deadlock)

Virtual Machine DoS (Crash or Deadlock)

Virtual Machine Escape (Other Guest)

Host DoS (Kernel Panic or Deadlock)

Virtual Machine Escape (Host)
Motivation

Hypervisor

VM 1
VM 2

Malicious Guest

(Privileged; Running in Ring-0)
Motivation

Hypervisor

VM 1

VM 2

Malicious Guest (Privileged; Running in Ring-0)

Local VM DoS (Crash or Deadlock)

Virtual Machine DoS (Crash or Deadlock)

Virtual Machine Escape (Other Guest)

Host DoS (Kernel Panic or Deadlock)

Virtual Machine Escape (Host)
Motivation

- **Malicious Guest** (Privileged; Running in Ring-0)
- **Local VM DoS** (Crash or Deadlock)
- **Virtual Machine DoS** (Crash or Deadlock)
- **Virtual Machine Escape** (Other Guest)
- **Host DoS** (Kernel Panic or Deadlock)
- **Virtual Machine Escape** (Host)
Motivation

Hypervisor

VM 1

VM 2

Virtual Machine DoS
(Crash or Deadlock)
Motivation

Hypervisor

VM 1

VM 2

Virtual Machine Escape
(Other Guest)
Motivation

Hypervisor

VM 1

VM 2

Malicious Guest
(Privileged; Running in Ring-0)

Local VM DoS (Crash or Deadlock)

Virtual Machine DoS (Crash or Deadlock)

Virtual Machine Escape (Other Guest)

Host DoS (Kernel Panic or Deadlock)

Virtual Machine Escape (Host)
Motivation

Malicious Guest (Privileged; Running in Ring-0)

Virtual Machine DoS (Crash or Deadlock)

Virtual Machine Escape (Other Guest)

Host DoS (Kernel Panic or Deadlock)
Motivation

VM 1 VM 2

Hypervisor

VM 1

VM 2

Virtual Machine Escape (Host)
Motivation

<table>
<thead>
<tr>
<th>Program Name</th>
<th>Eligible Entries</th>
<th>Bounty Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Hyper-V</td>
<td>Critical remote code execution, information disclosure and denial of services vulnerabilities</td>
<td>Up to $250,000 USD</td>
</tr>
<tr>
<td>Windows Defender Application Guard</td>
<td>Critical vulnerabilities in Windows Defender Application Guard</td>
<td>Up to $30,000 USD</td>
</tr>
<tr>
<td>Microsoft Edge (Chromium-based)</td>
<td>Critical and important vulnerabilities in Microsoft Edge (Chromium-based)</td>
<td>Up to $30,000</td>
</tr>
<tr>
<td>Office Insider</td>
<td>Vulnerabilities on Office Insider</td>
<td>Up to $15,000 USD</td>
</tr>
</tbody>
</table>
Challenge

Fuzzer of your Choice
Challenge

Fuzzer of your Choice → Target Software
Challenge

Fuzzer of your Choice

Target Software
Challenge

User Space Fuzzing
Hypervisor Fuzzing
Attack Surface
Hypervisor Attack Surface

- Guest
- Hypervisor

- Privileged Instructions
- Mov cr4, 0xAF
- ...
Hypervisor Attack Surface

Trap and Emulate

Guest

Hypervisor

• Memory-Mapped I/O (MMIO)
• Legacy Port I/O (PIO)
• Direct Memory Access (DMA)
• Hypercalls
• ...

① Emulation Request
② Return to Guest

Trap and Emulate
VM Exit
Hypervisor Attack Surface

Privileged Instructions

Guest

Hypervisor

Code

\[ \text{mov cr4, 0xAF} \]

\[ \ldots \]

\[ \ldots \]

Emulation Request

Return to Guest

• Memory-Mapped I/O (MMIO)
• Legacy Port I/O (PIO)
• Direct Memory Access (DMA)
• Hypercalls
Hypervisor Attack Surface

- Memory-Mapped I/O (**MMIO**)  
- Legacy Port I/O (**PIO**)  
- Hypercalls  
- Direct Memory Access (**DMA**)  
- ...
Implementation
Design Goals

• x86 Hypervisor Agnostic
• Blackbox Fuzzing with High Throughput
• High-Dimensional in Terms of
  ➤ Interfaces
  ➤ Operations
Our Approach

Hypervisor
Design Goals

• Blackbox Fuzzing with High Throughput
• High-Dimensional in Terms of
  ➤ Interfaces
  ➤ Operations

Our Approach

Hypervisor

VM

PCI Devices
ISA Devices
HPET
PIC
APIC
Chipset
MSR
Hypercalls

Tesseract Interpreter
Design Goals

• Blackbox Fuzzing with High Throughput
• High-Dimensional in Terms of
  ➤ Interfaces
  ➤ Operations

x86 Hypervisor Agnostic

Our Approach

Hypervisor

VM

Hyper-Cube OS

Tesseract Interpreter

PCI Devices
ISA Devices
HPET
PIC
APIC
Chipset
MSR
Hypercalls
Design Goals
• Blackbox Fuzzing with High Throughput
• High-Dimensional in Terms of
  ➤ Interfaces
  ➤ Operations

Our Approach

Hypervisor

VM

Hyper-Cube OS

Interface Enumeration

PCI Devices
ISA Devices
HPET
PIC
APIC
Chipset
MSR
Hypercalls

Tesseract Interpreter
Our Approach

Design Goals

• Blackbox Fuzzing with High Throughput
• High-Dimensional in Terms of
  ➤ Interfaces
  ➤ Operations

• x86 Hypervisor Agnostic

Our Approach

Hypervisor

VM

Hyper-Cube OS

Interface
Enumeration

PCI Devices
ISA Devices
MSR
Hypercalls
PIC
HPET
APIC
Chipset

Tesseract
Interpreter
Our Approach

Hypervisor

VM

Hyper-Cube OS

Tesseract Interpreter

PCI Devices
ISA Devices
MSR
Hypercalls
PIC
HPET
APIC
Chipset
Tesseract Handlers

- memset_mmio
- writes_io
- write_io
- reads_io
- write_mmio
- memset_io
- read_mmio
- reads_mmio
- write_msr
- kvm_hypercall
- xor_io
- read_io
- xor_mmio
- writes_mmio
- io_write_scratch_ptr
- bruteforce_mmio
- vmport
- mmio_write_scratch_ptr
- bruteforce_io
0120: 2fff 1c27 ab47 5700
0128: adf2 3d60 092f 5488
0130: ec2d 9d1a 029d 56fd
0138: e0d1 a275 1f56 1d28
0140: ea78 a2fa db07 d60d
0148: 1288 3a5a 91f9 1756
0150: 1cae 31ad 9b9c 938e
0158: 2a33 f597 6615 e267
0160: 0117 1f16 b440 8a86
0168: 9154 5b55 e4ca 9e3d
0170: 9d19 ae79 efac e500
0178: 8cdf 8c00 9a83 df76
0180: 91fe d779 026c 2e2b
0188: 9137 1ef8 eea3 d29c
0190: 1789 5938 a36f 718a
0198: 81e4 678c 20f5 fa0b
01a0: 774d 07f1 cee3 62bc
01a8: d845 bc86 7631 6eac

PRNG Stream

...
Tesseract Interpreter

PRNG Stream

0120: 2fff 1c27 ab47 5700
0128: adf2 3d60 092f 5488
0130: ec2d 9d1a 029d 56fd
0138: e0d1 a275 1f56 1d28
0140: ea78 a2fa db07 d60d
0148: 20bb 3f5a 91f9 1756
0150: 1ca3 31ad 9b9c 938e
0158: 2a33 f597 6615 e267
0160: 0117 1f16 b440 8a86
0168: 9154 5b55 e4ca 9e3d
0170: 9d19 ae79 efac e500
0178: 8cde 8c00 9a83 df76
0180: 91fe d799 026c 2e2b
0188: 9137 1789 5938 a36f 718a
0190: 1789 5938 a36f 718a
0198: 81e4 678c 20f5 fa0b
01a0: 774d 07f1 cee3 62bc
01a8: d845 bc86 7631 6eac

Opcode Handler

- **vmport** (0xbd4, 0x10ea)
- **memset_io** (0x426, 0xce0, 0x9dc, 0xca8)
- **writes_mmio** (0xec8, 0xad, 0x10ac, 0x7e9)
- **bruteforce_mmio** (0xce4, 0xdfa, 0xe31, 0x322)
- **writes_io** (0x4bb, 0xb8, 0xeb1, 0x401)
- **memset_mmio** (0x128, 0xa73, 0x2b3, 0xa84)
- **read_mmio** (0xbf3, 0x907)
- **bruteforce_io** (0x5c4, 0x49a, 0x94f, 0xb1c)
- **xor_mmio** (0x54b, 0xa00, 0xb51)
Evaluation

Tested Hypervisors
- KVM/QEMU
- Intel ACRN
- VMware Fusion
- Parallels Desktop (14.1.3)
- FreeBSD bhyve
- VirtualBox (12.0-RELEASE)
- (5.1.37_Ubuntu r122592)
- (4.0.1-rc4)
- (29360 Build)
- (11.0.3)

Results
- Assert Failures: 25
- Null-Pointer Dereferences: 13
- Memory-Corruptions: 8
- Div-By-Zero (FP Exceptions): 5
- Deadlocks: 4

Case Study: bhyve
- CVE-2019-12071: FreeBSD Kernel Denial of Service via Privileged Guest

CVE Rediscovery
- CVE-2015-3456: VENOM Vulnerability

TCG Mode:
- 5.8 sec (average time in seconds over 20 runs each)

KVM Mode:
- 49.7 sec

Hyper-Cube vs. VDF
- VDF: Targeted Evolutionary Fuzz Testing of Virtual Devices
- RAID 2017: Research in Attacks, Intrusions, and Defenses
- • AFL-based Fuzzing Approach
- • Fuzzing of Specific Device Emulators

Fuzzing 15 Device Emulators (QEMU-2.5.0)
- VDF: 13 /15 More Coverage
- HYPER CUBE: 2/15 More Coverage
- 10 Minutes Each
- ≈ 60 Days Each
Tested Hypervisors

FreeBSD bhyve (12.0-RELEASE)
VirtualBox (5.1.37_Ubuntu r122592)
Parallels Desktop (14.1.3)
KVM/QEMU (4.0.1-rc4)
Intel ACRN (29360 Build)
VMware Fusion (11.0.3)
# Results

<table>
<thead>
<tr>
<th>Category</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assert Failures</td>
<td>25</td>
</tr>
<tr>
<td>Null-Pointer Dereferences</td>
<td>13</td>
</tr>
<tr>
<td>Memory-Corruptions</td>
<td>8</td>
</tr>
<tr>
<td>Div-By-Zero (FP Exceptions)</td>
<td>5</td>
</tr>
<tr>
<td>Deadlocks</td>
<td>4</td>
</tr>
</tbody>
</table>

55 Bugs

---

**Case Study: bhyve**

CVE-2019-12071
FreeBSD Kernel Denial of Service via Privileged Guest

CVE Rediscovery

---

**TCG Mode:**

<table>
<thead>
<tr>
<th>Mode</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>KVM</td>
<td>49.7</td>
</tr>
<tr>
<td>TCG</td>
<td>5.8</td>
</tr>
</tbody>
</table>

(average time in seconds over 20 runs each)

---

**Hyper-Cube vs. VDF**

Fuzzing 15 Device Emulators (QEMU-2.5.0)

<table>
<thead>
<tr>
<th></th>
<th>Coverage</th>
<th>Crashes</th>
</tr>
</thead>
<tbody>
<tr>
<td>HYPER</td>
<td>13/15</td>
<td>4/15</td>
</tr>
<tr>
<td>CUBE</td>
<td>2/15</td>
<td></td>
</tr>
</tbody>
</table>

10 Minutes Each ≈ 60 Days Each

---

**VDF**: Targeted Evolutionary Fuzz Testing of Virtual Devices

RAID 2017: Research in Attacks, Intrusions, and Defenses

- AFL-based Fuzzing Approach
- Fuzzing of Specific Device Emulators
Case Study: bhyve

CVE-2019-12071
FreeBSD Kernel Denial of Service via Privileged Guest
Case Study: bhyve

CVE-2019-12071
FreeBSD Kernel Denial of Service via Privileged Guest

-------------- INTERPRETER CONFIGURATION --------------

mmio_area[0] = {
    base = 0xfee00000;
    size = 0x00008000;
    desc = "APIC";
};

-------------- INTERPRETER EXECUTING ... --------------

mmio_memset_32(0x00000c7a + mmio_area[0], 0x884f972f, 0x0000001b)

...
Case Study: bhyve

CVE-2019-12071
FreeBSD Kernel Denial of Service via Privileged Guest

Translates to

```
mnio_memset_32:
  lea    edi, [APIC_addr+offset]
  mov    esi, payload
  mov    ecx, n
  rep movsd
```

mmio_memset_32(0x00007c7a + mmio_area[0], 0x884f972f, 0x0000001b)
Case Study: bhyve

CVE-2019-12071
FreeBSD Kernel Denial of Service via Privileged Guest

TCG Mode: 5.8 sec (average time in seconds over 20 runs each)
KVM Mode: 49.7 sec

Hyper-Cube vs. VDF
VDF: Targeted Evolutionary Fuzz Testing of Virtual Devices
RAID 2017: Research in Attacks, Intrusions, and Defenses

• AFL-based Fuzzing Approach
• Fuzzing of Specific Device Emulators

Fuzzing 15 Device Emulators (QEMU-2.5.0)

VDF
HYPER
CUBE
13 /15 More Coverage
2 /15 More Coverage
9 /15 Crashed
4 /15 Crashed
10 Minutes Each
≈ 60 Days Each
Case Study: bhyve

CVE-2019-12071
FreeBSD Kernel Denial of Service via Privileged Guest

panic: emulate_movs: unexpected error 22
Evaluated Hypervisors

- KVM/QEMU
- Intel ACRN
- VMware Fusion
- Parallels Desktop (14.1.3)
- FreeBSD bhyve
- VirtualBox (12.0-RELEASE)
  (5.1.37_Ubuntu r122592)
  (4.0.1-rc4)
  (29360 Build)
  (11.0.3)

Results

<table>
<thead>
<tr>
<th>Issue Type</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assert Failures</td>
<td>25</td>
</tr>
<tr>
<td>Null-Pointer Dereferences</td>
<td>13</td>
</tr>
<tr>
<td>Memory-Corruptions</td>
<td>8</td>
</tr>
<tr>
<td>Div-By-Zero (FP Exceptions)</td>
<td>5</td>
</tr>
<tr>
<td>Deadlocks</td>
<td>4</td>
</tr>
</tbody>
</table>

CVE Rediscovery

- CVE-2015-3456
  VENOM Vulnerability

Case Study: bhyve

CVE-2019-12071
FreeBSD Kernel Denial of Service via Privileged Guest

TCG Mode: 5.8 sec
(average time in seconds over 20 runs each)

KVM Mode: 49.7 sec
(average time in seconds over 20 runs each)

Hyper-Cube vs. VDF

- Fuzzing 15 Device Emulators (QEMU-2.5.0)
  - VDF: Targeted Evolutionary Fuzz Testing of Virtual Devices
  - RAID 2017: Research in Attacks, Intrusions, and Defenses
    - AFL-based Fuzzing Approach
    - Fuzzing of Specific Device Emulators

VDF: 13/15 More Coverage
HYPER: 2/15 More Coverage

- 9/15 Crashed
- 4/15 Crashed

10 Minutes Each
≈ 60 Days Each
Tested Hypervisors

- KVM/QEMU
- Intel ACRN
- VMware Fusion
- Parallels Desktop (14.1.3)
- FreeBSD bhyve
- VirtualBox (12.0-RELEASE)
  (5.1.37_Ubuntu r122592)
  (4.0.1-rc4)
  (29360 Build)
  (11.0.3)

Results

<table>
<thead>
<tr>
<th>Issue Type</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assert Failures</td>
<td>25</td>
</tr>
<tr>
<td>Null-Pointer Dereferences</td>
<td>13</td>
</tr>
<tr>
<td>Memory-Corruptions</td>
<td>8</td>
</tr>
<tr>
<td>Div-By-Zero (FP Exceptions)</td>
<td>5</td>
</tr>
<tr>
<td>Deadlocks</td>
<td>4</td>
</tr>
</tbody>
</table>

CVE Rediscovery

CVE-2015-3456
VENOM Vulnerability

TCG Mode: 5.8 sec
(average time in seconds over 20 runs each)

Hyper-Cube vs. VDF

- VDF: Targeted Evolutionary Fuzz Testing of Virtual Devices
- RAID 2017: Research in Attacks, Intrusions, and Defenses
- AFL-based Fuzzing Approach
- Fuzzing of Specific Device Emulators

Fuzzing 15 Device Emulators (QEMU-2.5.0)

- HYPERCUBE: 13/15 More Coverage
- VDF: 2/15 More Coverage

- 9/15 Crashed
- 4/15 Crashed

10 Minutes Each ≈ 60 Days Each
Tested Hypervisors
- KVM/QEMU
- Intel ACRN
- VMware Fusion
- Parallels Desktop (14.1.3)
- FreeBSD bhyve
- VirtualBox (12.0-RELEASE)
- (5.1.37_Ubuntu r122592)
- (4.0.1-rc4)
- (29360 Build)
- (11.0.3)

Results
- Assert Failures: 25
- Null-Pointer Dereferences: 13
- Memory-Corruptions: 8
- Div-By-Zero (FP Exceptions): 5
- Deadlocks: 4
- Bugs: 55

Case Study: bhyve
- CVE-2019-12071: FreeBSD Kernel Denial of Service via Privileged Guest

CVE Rediscovery
- CVE-2015-3456: VENOM Vulnerability

TCG Mode: 5.8 sec
KVM Mode: 49.7 sec

Fuzzing 15 Device Emulators (QEMU-2.5.0)

• AFL-based Fuzzing Approach
• Fuzzing of Specific Device Emulators

Hyper-Cube vs. VDF
- VDF: Targeted Evolutionary Fuzz Testing of Virtual Devices
- RAID 2017: Research in Attacks, Intrusions, and Defenses
- 13/15 More Coverage
- 9/15 Crashed
- ≈60 Days Each

• HYPER
• CUBE

10 Minutes Each
VDF: Targeted Evolutionary Fuzz Testing of Virtual Devices

RAID 2017: Research in Attacks, Intrusions, and Defenses

- AFL-based Fuzzing Approach
- Fuzzing of Specific Device Emulators
Hyper-Cube vs. VDF

Fuzzing 15 Device Emulators (QEMU-2.5.0)
### Tested Hypervisors

- KVM/QEMU
- Intel ACRN
- VMware Fusion
- Parallels Desktop (14.1.3)
- FreeBSD bhyve
- VirtualBox (12.0-RELEASE) (5.1.37_Ubuntu r122592) (4.0.1-rc4) (29360 Build) (11.0.3)

### Results

- **Assert Failures**: 25
- **Null-Pointer Dereferences**: 13
- **Memory-Corruptions**: 8
- **Div-By-Zero (FP Exceptions)**: 5
- **Deadlocks**: 4

**Bugs**: 55

### Case Study: bhyve

- **CVE-2019-12071**: FreeBSD Kernel Denial of Service via Privileged Guest

### Hyper-Cube vs. VDF

**Fuzzing 15 Device Emulators** (QEMU-2.5.0)

<table>
<thead>
<tr>
<th></th>
<th>HYPER CUBE</th>
<th>VDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>More Coverage</td>
<td>13/15</td>
<td>2/15</td>
</tr>
<tr>
<td>Crashed</td>
<td>9/15</td>
<td>4/15</td>
</tr>
</tbody>
</table>

- **TCG Mode**: 5.8 sec (average time in seconds over 20 runs each)
- **KVM Mode**: 49.7 sec

**Fuzzing 15 Device Emulators (QEMU-2.5.0)**

**Hyper-Cube** vs. **VDF**

- **VDF**
- **HYPER CUBE**

- **VDF**: Targeted Evolutionary Fuzz Testing of Virtual Devices
- **RAID 2017**: Research in Attacks, Intrusions, and Defenses
- **Fuzzing 15 Device Emulators (QEMU-2.5.0)**

- **VDF** vs. **HYPER CUBE**
  - 15 Device Emulators
  - **Fuzzing Approach**: AFL-based
  - **Fuzzing Specific Device Emulators**

- **Coverage**
  - **HYPER CUBE**: 13/15 More Coverage
  - **VDF**: 2/15 More Coverage

- **Crashed**
  - **HYPER CUBE**: 9/15
  - **VDF**: 4/15

- **Time**
  - **10 Minutes Each**
  - **Approximately 60 Days Each**
Fuzzing 15 Device Emulators (QEMU-2.5.0)

**Hyper-Cube** vs. **VDF**

**Hyper-Cube**
- \(13/15\) More Coverage
- \(9/15\) Crashed

**VDF**
- \(2/15\) More Coverage
- \(4/15\) Crashed
Hyper-Cube vs. VDF

Fuzzing 15 Device Emulators (QEMU-2.5.0)

**Hyper-Cube**

- 13/15 More Coverage
- 9/15 Crashed
- 10 Minutes Each

**VDF**

- 2/15 More Coverage
- 4/15 Crashed
- ≈ 60 Days Each
Conclusion
• **Novel Technique** to Fuzz Hypervisors

• **Outperforms** Coverage-Guided Fuzzers

• **Full-System** Fuzzing