A Non-Inclusive Memory Permissions Architecture for Protection Against Cross-Layer Attacks

Jesse Elwell\textsuperscript{1} Ryan Riley\textsuperscript{2}
Nael Abu-Ghazaleh\textsuperscript{1} Dmitry Ponomarev\textsuperscript{1}

\textsuperscript{1}State University of New York at Binghamton
Department of Computer Science

\textsuperscript{2}Qatar University
Department of Computer Science

20th International Symposium on High Performance Computer Architecture
February 17th, 2014
Introduction & Motivation

- System software (Hypervisor/OS) is steadily increasing in complexity
- Complexity leads to vulnerabilities

<table>
<thead>
<tr>
<th>Software</th>
<th>Lines of Code</th>
<th>Vulnerabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>KVM</td>
<td>30K</td>
<td>38</td>
</tr>
<tr>
<td>Xen</td>
<td>200K</td>
<td>59</td>
</tr>
<tr>
<td>Linux kernel</td>
<td>15M</td>
<td>228</td>
</tr>
</tbody>
</table>

A single vulnerability in system software can allow an attacker to compromise the entire system
Example 1: Malicious Supervisor Attack

### x86-64 Memory Permissions

<table>
<thead>
<tr>
<th>EXECUTABLE</th>
<th>SUPERVISOR OR USER/SUPERVISOR</th>
<th>READ-ONLY OR READ-WRITE</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES/NO</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Memory Layout

- **OS**
- **User**
Example 1: Malicious Supervisor Attack

**x86-64 Memory Permissions**

<table>
<thead>
<tr>
<th>EXECUTABLE</th>
<th>SUPERVISOR OR USER/SUPERVISOR</th>
<th>READ-ONLY OR READ-WRITE</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES/NO</td>
<td>USER/SUPERVISOR</td>
<td>READ-WRITE</td>
</tr>
<tr>
<td>NO</td>
<td>USER/SUPERVISOR</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Memory Layout:

- **OS**
- **User**
- **Sensitive Data**
Example 1: Malicious Supervisor Attack

### x86-64 Memory Permissions

<table>
<thead>
<tr>
<th>EXECUTABLE YES/NO</th>
<th>SUPERVISOR OR USER/SUPERVISOR</th>
<th>READ-ONLY OR READ-WRITE</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>USER/SUPERVISOR</td>
<td>READ-WRITE</td>
</tr>
</tbody>
</table>

#### Memory Layout

- Buffer
- OS
- User
- Sensitive Data
Example 1: Malicious Supervisor Attack

### x86-64 Memory Permissions

<table>
<thead>
<tr>
<th>EXECUTABLE</th>
<th>SUPERVISOR OR USER/SUPERVISOR</th>
<th>READ-ONLY OR READ-WRITE</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES/NO</td>
<td>USER/SUPERVISOR</td>
<td>READ-WRITE</td>
</tr>
<tr>
<td>NO</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Memory Layout:
- **Buffer**
- **Copy**
- **Sensitive Data**

---

Binghamton University / Qatar University  
HPCA 2014
Example 1: Malicious Supervisor Attack

### x86-64 Memory Permissions

<table>
<thead>
<tr>
<th>EXECUTABLE</th>
<th>SUPERVISOR OR USER/SUPERVISOR</th>
<th>READ-ONLY OR READ-WRITE</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>USER/SUPERVISOR</td>
<td>READ-WRITE</td>
</tr>
</tbody>
</table>

#### Memory Layout

- **Sensitive Data**
- **OS**
- **User**
- **Sensitive Data**
Example 1: Malicious Supervisor Attack

### x86-64 Memory Permissions

<table>
<thead>
<tr>
<th>EXECUTABLE</th>
<th>SUPERVISOR OR USER/SUPERVISOR</th>
<th>READ-ONLY OR READ-WRITE</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES/NO</td>
<td>USER/SUPERVISOR</td>
<td>READ-WRITE</td>
</tr>
</tbody>
</table>

#### Memory Layout

- **Sensitive Data**
- **OS**
- **User**
- **Sensitive Data**
Example 2: return-2-user Attack

### x86-64 Memory Permissions

<table>
<thead>
<tr>
<th>EXECUTABLE</th>
<th>SUPERVISOR OR USER/SUPERVISOR</th>
<th>READ-ONLY OR READ-WRITE</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES/NO</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Memory Layout

- **OS**
- **User**
### Example 2: return-2-user Attack

#### x86-64 Memory Permissions

<table>
<thead>
<tr>
<th>EXECUTABLE</th>
<th>SUPERVISOR OR USER/SUPERVISOR</th>
<th>READ-ONLY OR READ-WRITE</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES/NO</td>
<td>USER/SUPERVISOR</td>
<td>READ-WRITE</td>
</tr>
</tbody>
</table>

#### Memory Layout

```
<table>
<thead>
<tr>
<th>OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
</tr>
<tr>
<td>Malicious Code</td>
</tr>
</tbody>
</table>
```

---

Note: The diagram shows a section labeled "Malicious Code."
### x86-64 Memory Permissions

<table>
<thead>
<tr>
<th>EXECUTABLE</th>
<th>SUPERVISOR OR USER/SUPERVISOR</th>
<th>READ-ONLY OR READ-WRITE</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES/NO</td>
<td>USER/SUPERVISOR</td>
<td>READ-WRITE</td>
</tr>
</tbody>
</table>

#### Memory Layout

- Code
- OS
- User
- System Call
- Malicious Code
### x86-64 Memory Permissions

<table>
<thead>
<tr>
<th>EXECUTABLE</th>
<th>SUPERVISOR OR USER/SUPERVISOR</th>
<th>READ-ONLY OR READ-WRITE</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES/NO</td>
<td>USER/SUPERVISOR</td>
<td>READ-WRITE</td>
</tr>
</tbody>
</table>

**Example 2: return-2-user Attack**
Example 2: return-2-user Attack

### x86-64 Memory Permissions

<table>
<thead>
<tr>
<th>EXECUTABLE</th>
<th>SUPERVISOR OR USER/SUPERVISOR</th>
<th>READ-ONLY OR READ-WRITE</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES/NO</td>
<td>USER/SUPERVISOR</td>
<td>READ-WRITE</td>
</tr>
</tbody>
</table>

Memory Layout

- Code
- OS
- User
- Malicious Code

---

Vulnerability Exploited

System Call

OS Privileges

---

OS Privileges

---

OS Privileges

---

OS Privileges

---

OS Privileges

---

OS Privileges

---

OS Privileges

---

OS Privileges
Example 2: return-2-user Attack

### x86-64 Memory Permissions

<table>
<thead>
<tr>
<th>EXECUTABLE</th>
<th>SUPERVISOR OR USER/SUPERVISOR</th>
<th>READ-ONLY OR READ-WRITE</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES/NO</td>
<td>USER/SUPERVISOR</td>
<td>READ-WRITE</td>
</tr>
</tbody>
</table>

Memory Layout

- Code
- OS Privileges
- User
- Malicious Code

OS Privileges

- Code
- OS
- User
- Malicious Code

Binghamton University / Qatar University

HPCA 2014
Cross-Layer Attack Flows

App

Guest OS

Hypervisor

App

Guest OS

App

Guest OS
Cross-Layer Attack Flows

Hypervisor

App

Guest OS

ret-2-user

App

Guest OS

Hypervisor
Cross-Layer Attack Flows

App
ret-2-user
Guest OS
ret-2-VM
Hypervisor
App
Guest OS
App
Cross-Layer Attack Flows

- App
  - ret-2-user
  - Guest OS

- App
  - ret-2-VM
  - Hypervisor

- App
  - ret-2-user
  - Guest OS
Cross-Layer Attack Flows

Hypervisor

Guest OS

App

ret-2-user

ret-2-VM

App

Guest OS

App

Guest OS

ret-2-user

ret-2-VM
Cross-Layer Attack Flows

Hypervisor

App

Guest OS

App

Guest OS

ret-2-user

ret-2-VM

Hypervisor
Non-Inclusive Memory Permissions

Current Inclusive x86-64 Memory Permissions

<table>
<thead>
<tr>
<th>EXECUTABLE</th>
<th>SUPERVISOR OR USER/SUPERVISOR</th>
<th>READ-ONLY OR READ-WRITE</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES/NO</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Non-Inclusive Memory Permissions

Current Inclusive x86-64 Memory Permissions

<table>
<thead>
<tr>
<th>EXECUTABLE</th>
<th>SUPERVISOR OR USER/SUPERVISOR</th>
<th>READ-ONLY OR READ-WRITE</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES/NO</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Binghamton University / Qatar University

HPCA 2014
Non-Inclusive Memory Permissions

Non-Inclusive Memory Permissions (NIMP)

<table>
<thead>
<tr>
<th>Hypervisor</th>
<th>Operating System</th>
<th>User-Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>Write</td>
<td>Execute</td>
</tr>
<tr>
<td>Read</td>
<td>Write</td>
<td>Execute</td>
</tr>
<tr>
<td>Read</td>
<td>Write</td>
<td>Execute</td>
</tr>
</tbody>
</table>
Mitigating Malicious Supervisor Attacks

Non-Inclusive Memory Permissions

<table>
<thead>
<tr>
<th>Operating System</th>
<th>User-Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>Write</td>
</tr>
<tr>
<td>NO</td>
<td>NO</td>
</tr>
</tbody>
</table>

Memory Layout

- Buffer
- Copy
- Sensitive Data

Binghamton University / Qatar University
Mitigating Malicious Supervisor Attacks

Non-Inclusive Memory Permissions

<table>
<thead>
<tr>
<th>Operating System</th>
<th>User-Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>Write</td>
</tr>
<tr>
<td>NO</td>
<td>NO</td>
</tr>
</tbody>
</table>

Memory Layout

- Buffer
- Sensitive Data

EXCEPTION!
Mitigating Malicious Supervisor Attacks

Non-Inclusive Memory Permissions

Operating System
<table>
<thead>
<tr>
<th>Read</th>
<th>Write</th>
<th>Execute</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
</tbody>
</table>

User-Level
<table>
<thead>
<tr>
<th>Read</th>
<th>Write</th>
<th>Execute</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
</tbody>
</table>

Memory Layout
- Buffer
- Sensitive Data

EXCEPTION!
Mitigating return-2-user Attacks

### Non-Inclusive Memory Permissions

<table>
<thead>
<tr>
<th>Operating System</th>
<th>User-Level</th>
<th>Memory Layout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>Write</td>
<td>Execute</td>
</tr>
<tr>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
</tbody>
</table>

Non-Inclusive Memory Permissions

- **OS Privileges**
  - Code
  - OS
  - User
  - Malicious Code

**EXCEPTION**!
Mitigating return-2-user Attacks

Non-Inclusive Memory Permissions

<table>
<thead>
<tr>
<th>Operating System</th>
<th>User-Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>Write</td>
</tr>
<tr>
<td>NO</td>
<td>NO</td>
</tr>
</tbody>
</table>

Memory Layout

- Code
- EXCEPTION!
- Malicious Code

Binghamton University / Qatar University HPCA 2014
Mitigating return-2-user Attacks

Non-Inclusive Memory Permissions

<table>
<thead>
<tr>
<th>Operating System</th>
<th>User-Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>Write</td>
</tr>
<tr>
<td>NO</td>
<td>NO</td>
</tr>
</tbody>
</table>

Memory Layout

- Code
- Exception
- Malicious Code
NIMP Design Overview

Memory Permission Change Requests

Memory Permission Manager

Permission Store
NIMP Design Overview

Memory Permission Change Requests → Memory Permission Manager → Permission Store → Permission Reference Monitor

Memory Access Requests → Memory Access Decision
## The Permission Store

<table>
<thead>
<tr>
<th>15</th>
<th>14</th>
<th>13</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S</td>
<td>P</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Reserved**
- **Hypervisor**
- **OS**
- **User**

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>W</th>
<th>X</th>
<th></th>
<th>R</th>
<th>W</th>
<th>X</th>
<th></th>
<th>R</th>
<th>W</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td>11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Physical Memory

- **PS Entry 0**
- **PS Entry 1**
- **PS Entry N**
- **PS Entry 2**

### Permission Store (Protected Memory)
The Permission Store

- Physical Memory
- Permission Store (Protected Memory)
- Reserved
- Hypervisor
- OS
- User

- PS Entry 0
- PS Entry 1
- PS Entry 2
- PS Entry N
The Permission Store

<table>
<thead>
<tr>
<th>Reserved</th>
<th>S</th>
<th>P</th>
<th>T</th>
<th>Hypervisor</th>
<th>OS</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R</td>
<td>W</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R</td>
<td>W</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R</td>
<td>W</td>
<td>X</td>
</tr>
</tbody>
</table>

| PS_ENTRY 0 |
| PS_ENTRY 1 |
| PS_ENTRY 2 |
| PS_ENTRY N |

Physical Memory

Register

PS_BASE

PS Entry

Permission Store (Protected Memory)

Binghamton University / Qatar University

HPCA 2014

10 / 25
Augmenting TLBs to Store PS Entries

<table>
<thead>
<tr>
<th>Virtual Address</th>
<th>Physical Address</th>
<th>Virtual Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x12345000</td>
<td>0x09ABC000</td>
<td>NX U RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TLB

Binghamton University / Qatar University

HPCA 2014
Augmenting TLBs to Store PS Entries

<table>
<thead>
<tr>
<th>Virtual Address</th>
<th>Physical Address</th>
<th>Virtual Permissions</th>
<th>Permission Store Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x12345000</td>
<td>0x09ABC000</td>
<td>NX U RO</td>
<td>RW - RW - RW -</td>
</tr>
</tbody>
</table>

::: ::: :::
The Memory Permission Manager

Rule
Database

Current Permissions
Requester
New Permissions

Allow or Disallow
The Memory Permission Manager

Permission Change Request

Rule Database

Current Permissions Requester New Permissions

Allow or Disallow
The Memory Permission Manager

Rule Database

Current Permissions
Requester
New Permissions

Binghamton University / Qatar University  HPCA 2014  12 / 25
The Memory Permission Manager

Rule Database

Current Permissions
Requester
New Permissions

Allow or Disallow
## Contents of the Rule Database

<table>
<thead>
<tr>
<th>Requester</th>
<th>Initial Permissions</th>
<th>New Permissions</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hyp.</td>
<td>OS</td>
<td>User</td>
</tr>
<tr>
<td></td>
<td>R W X</td>
<td>R W X</td>
<td>R W X</td>
</tr>
<tr>
<td>Hypervisor</td>
<td>- - -</td>
<td>- - -</td>
<td>- - -</td>
</tr>
<tr>
<td>Hypervisor</td>
<td>* * *</td>
<td>* * *</td>
<td>* * *</td>
</tr>
<tr>
<td>OS</td>
<td>- - -</td>
<td>- - -</td>
<td>- - -</td>
</tr>
<tr>
<td>OS</td>
<td>- - -</td>
<td>* * *</td>
<td>* * *</td>
</tr>
<tr>
<td>Hypervisor</td>
<td>- W -</td>
<td>- - -</td>
<td>- - -</td>
</tr>
<tr>
<td>OS</td>
<td>- - -</td>
<td>- W -</td>
<td>- - -</td>
</tr>
<tr>
<td>OS</td>
<td>- - -</td>
<td>- - -</td>
<td>- W -</td>
</tr>
</tbody>
</table>
Secure Permission Changes: The PERM_SET Instruction

<table>
<thead>
<tr>
<th>Virtual Address</th>
<th>New Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>PERM_SET %eax,</td>
<td>%ebx</td>
</tr>
</tbody>
</table>
Secure Permission Changes: The PERM_SET Instruction

```
PERM_SET
\%
\%eax, \%ebx
Access
TLB
```

```
Virtual
Address
New
Permissions

Access
TLB
```
Secure Permission Changes: The PERM_SET Instruction

\[
\text{PERM\_SET} \quad \%\text{eax}, \quad \%\text{ebx}
\]

- Miss
- Access TLB
- Access Page Tables
- Read PS Entry
- Current Permissions
- Virtual Address
- New Permissions
- Access TLB Hit
- Exception No Match
- Perform Action Write PS + TLB Match

Binghamton University / Qatar University
HPCA 2014
14 / 25
Secure Permission Changes: The PERM_SET Instruction

```
PERM_SET
  %eax, %ebx
```

Virtual Address

New Permissions

Access TLB

Hit

Miss

Access Page Tables

Read PS Entry

Current Permissions

Requester (Current Privilege Level)
Secure Permission Changes: The PERM_SET Instruction
Secure Permission Changes: The PERM_SET Instruction

PERM_SET \( \%\text{eax}, \%\text{ebx} \)

- **Miss**
  - Access TLB
  - Access Page Tables
  - Read PS Entry
  - Current Permissions
  - Requester (Current Privilege Level)
  - Access Rule Database

- **Hit**
  - Virtual Address
  - New Permissions

Binghamton University / Qatar University  HPCA 2014  14 / 25
Secure Permission Changes: The PERM_SET Instruction

PERM_SET %eax, %ebx

Hit

Access TLB

Miss

Access Page Tables

Read PS Entry

Current Permissions

Requester (Current Privilege Level)

Match

Access Rule Database

Perform Action

Write PS + TLB

Virtual Address

New Permissions
Secure Permission Changes: The PERM_SET Instruction

**Flowchart Description:**
- **PERM_SET** (%eax, %ebx)
- **Virtual Address**
- **New Permissions**
- **Access TLB**
- **Hit**
- **Miss**
- **Access Page Tables**
- **Read PS Entry**
- **Current Permissions**
- **Requester (Current Privilege Level)**
- **Access Rule Database**
  - **Match**
  - **No Match**
  - **Perform Action**
  - **Write PS + TLB**
  - **Exception**
The Permission Reference Monitor

<table>
<thead>
<tr>
<th>Load/Store Instructions</th>
<th>Permission Store</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Expected Permissions</strong></td>
<td><strong>Actual Permissions</strong></td>
</tr>
</tbody>
</table>

- Expected permissions can be:
  - Embedded into instruction bits
  - Stored in a new register
The Permission Reference Monitor

- Expected permissions can be:
  - Embedded into instruction bits
  - Stored in a new register
The Permission Reference Monitor

- Expected permissions can be:
  - Embedded into instruction bits
  - Stored in a new register
Hardware Changes Needed for NIMP

![Diagram of hardware changes needed for NIMP]

- CPU
  - Core 0
    - PS Entries
    - ITLB
    - DTLB
  - Core 1
    - PS Entries
    - ITLB
    - DTLB
- MPM
- MMU
- Rule Database
- PS_Base Register
- PERM_SET
- Hypervisor
- OS
- Physical Memory
- Regular Memory
- Protected Memory
- PS Table
Hardware Changes Needed for NIMP
Hardware Changes Needed for NIMP
Hardware Changes Needed for NIMP

- CPU
  - Core 0
    - PS Entries
    - DTLB
  - Core 1
    - PS Entries
    - DTLB

- MPM
- MMU
- Rule Database
- PS_Base Register
- PERM_SET

- Physical Memory
- Regular Memory
- Protected Memory
- PS Table

- Hypervisor
- OS
Performance Evaluation: Sources of Overhead

- Fetching PS entries from the Permission Store on TLB misses
  - Cached in various levels of (data) caches
Performance Evaluation: Sources of Overhead

- Fetching PS entries from the Permission Store on TLB misses
  - Cached in various levels of (data) caches

- Cycles spent performing actions before permissions are changed
  - Zeroing pages
Performance Evaluation: Sources of Overhead

- Fetching PS entries from the Permission Store on TLB misses
  - Cached in various levels of (data) caches

- Cycles spent performing actions before permissions are changed
  - Zeroing pages

- Increase in cycle time due to hardware component delay
  - Widening TLB entries
  - Accessing the Rule Database
Performance Evaluation: Fetching PS Entries

- We used MARSSx86, a full system x86-64 simulator to evaluate the impact of NIMP on cache performance

- Overall effect on IPC

- Miss/hit rates for PS Entry data

- Effect on miss/hit rate for regular data
Reduction in IPC

- Calculix: 0.2%
- Gamess: 0.1%
- Gromacs: 0.3%
- Libquantum: 3.8%
- MCF: 3.5%
- Average: 1.3%
Reduction in IPC

- Calculix: 0.2%
- Gamess: 0.1%
- Gromacs: 0.3%
- Libquantum: 3.8%
- MCF: 3.5%
- Average: 1.3%
Reduction in IPC

- Calculix: 0.2%
- GameSS: 0.1%
- Gromacs: 0.3%
- Libquantum: 3.8%
- MCF: 3.5%
- Average: 1.3%
Reduction in IPC

![Bar chart showing IPC difference in percentages for various applications. The applications include calculix, gamess, gromacs, libquantum, mcf, and average. The highest difference is 3.8%, and the average difference is 1.3%.]
L1 Miss Rate Accessing Permission Bits

Miss Rate (%)
- Calculix: 0.77%
- GameSS: 0.67%
- Gromacs: 2.88%
- Libquantum: 19.70%
- MCF: 16.10%
- Average: 4.16%
L1 Miss Rate Accessing Permission Bits

- Calculix: 0.77%
- GameSS: 0.67%
- Gromacs: 2.88%
- Libquantum: 19.70%
- MCF: 16.10%
- Average: 4.16%
L1 Miss Rate Accessing Permission Bits

- Calculix: 0.77%
- Game: 0.67%
- Gromacs: 2.88%
- Libquantum: 19.70%
- MCF: 16.10%
- Average: 4.16%
L1 Miss Rate Accessing Permission Bits

- Calculix: 0.77%
- GameS: 0.67%
- Gromacs: 2.88%
- LibQuantum: 19.70%
- MCF: 16.10%
- Average: 4.16%
L1 Cache Miss Rate for Regular Accesses

Without Perms

With Perms

Δ 0.3%

Δ 0.4%

Δ 0.08%
L1 Cache Miss Rate for Regular Accesses

Without Perms  With Perms

Miss Rate (%)

Δ 0.3%

Δ 0.4%

Δ 0.08%
L1 Cache Miss Rate for Regular Accesses

![Chart showing L1 Cache Miss Rate for Regular Accesses with and without permissions, along with percentage changes for different benchmarks.]

- Without Perms
- With Perms

- Δ 0.3%
- Δ 0.4%
- Δ 0.08%
Performance Evaluation: Zeroing Pages

- We profiled the Linux kernel using *ftrace* to collect information about events that cause permission change requests.

- Assumptions:
  - Every permission transition requires the page to be zeroed.
  - 1 cycle / byte (i.e. 4096 cycles / 4KB page).
  - Cycle percentages assume a 3GHz Processor.
Page Zeroing Overhead

- **VirtualBox**
  - Booting a virtual machine
- **Chromium**
  - Loading web pages
- **LibreOffice**
  - Opening spreadsheets

<table>
<thead>
<tr>
<th>Application</th>
<th>Changes Per Second</th>
<th>Cycle Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>VirtualBox</td>
<td>2765</td>
<td>0.4%</td>
</tr>
<tr>
<td>Chromium</td>
<td>2973</td>
<td>0.4%</td>
</tr>
<tr>
<td>LibreOffice</td>
<td>8608</td>
<td>1.2%</td>
</tr>
</tbody>
</table>
Conclusions

- Vulnerabilities in system software coupled with inclusive memory permissions in current designs leave systems exposed to cross-layer attacks.

- Non-inclusive permissions can stop these attacks with minimal overhead.

- NIMP incurs about 1% performance loss on average, and modest changes to hardware and system software.
Thank you!  
Questions/Comments?