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

Jesse Elwell\textsuperscript{1} \quad Ryan Riley\textsuperscript{2}

Nael Abu-Ghazaleh\textsuperscript{1} \quad 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
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>
</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>
</tbody>
</table>

**Memory Layout**

- OS
- User
- Sensitive Data

---

Buffers Sensitive Data NO USER/SUPERVISOR READ-WRITE
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>
<th>Memory Layout</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>USER/SUPERVISOR</td>
<td>READ-WRITE</td>
<td>Buffer</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>User</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Sensitive Data</td>
</tr>
</tbody>
</table>
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

- Buffer
- Copy
- 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>
<th>Memory Layout</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES/NO</td>
<td>USER/SUPERVISOR</td>
<td>READ-WRITE</td>
<td></td>
</tr>
<tr>
<td>NO</td>
<td>USER/SUPERVISOR</td>
<td>READ-WRITE</td>
<td>Sensitive Data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>User</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Sensitive Data</td>
</tr>
</tbody>
</table>

Sensitive Data

Buffer Copy

Sensitive Data

Memory Layout
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**
- 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>
<th>Memory Layout</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES/NO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></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

- **OS**
- **User**
- **Malicious Code**
### 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
- System Call
- Malicious Code
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 Diagram](image_url)
### 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
- OS Privileges
- Malicious Code

OS privileges are critical in preventing unauthorized access to sensitive system resources. The diagram above illustrates how malware can exploit vulnerabilities in the memory layout and permissions to gain unauthorized access to the system. The malicious code is shown to have read-write access, allowing it to modify system resources and evade detection.
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>

- **EXECUTABLE**:
  - YES/NO

- **SUPERVISOR OR USER/SUPERVISOR**:
  - USER/SUPERVISOR

- **READ-ONLY OR READ-WRITE**:
  - READ-WRITE

#### Memory Layout

- **Code**
- **OS Privileges**
- **User**
- **Malicious Code**
Cross-Layer Attack Flows

App

App

Hypervisor

Guest OS

Guest OS

V M
Cross-Layer Attack Flows

![Diagram showing cross-layer attack flows involving a hypervisor, guest OS, and apps. Arrows indicate flow from app to guest OS and then to the hypervisor.]
Cross-Layer Attack Flows

App

Guest OS

ret-2-user

Hypervisor

ret-2-VM

Guest OS

App

App
Cross-Layer Attack Flows

```
ret-2-user
Guest OS
App
ret-2-VM
Hypervisor
Guest OS
App
ret-2-user
```

Binghamton University / Qatar University  HPCA 2014  5 / 25
Cross-Layer Attack Flows

Hypervisor

App

Guest OS

App

Guest OS

App

Guest OS

ret-2-user

ret-2-VM

Hypervisor
Cross-Layer Attack Flows

![Diagram showing cross-layer attack flows with arrows indicating ret-2-user and ret-2-VM]
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 (NIMP)
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

Non-Inclusive Memory Permissions (NIMP)

Hypervisor
- Read
- Write
- Execute

Operating System
- Read
- Write
- Execute

User-Level
- Read
- Write
- Execute
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
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

<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 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>

Memory Layout:
- Code
- User
- Malicious Code

OS Privileges:
- 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>
<tr>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>

Memory Layout

- Code
- Exception!
- 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
- User-Level
- Malicious Code

EXCEPTION!
NIMP Design Overview

Permission Store
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

Reserved

S

PT

Hypervisor

OS

User

R

W

X

R

W

X

R

W

X

Physical Memory

PS Entry 0

PS Entry 1

PS Entry N

PS Entry 2

0124 356789101112131415

User

OS

Hypervisor

P

T

S

X

W

R

X

W

R

X

W

R

Reserved

Permission Store (Protected Memory)
The Permission Store

Reserved

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

S P T

Hypervisor OS User

R W X R W X R W X

Permission Store (Protected Memory)

PS Entry 0
PS Entry 1
PS Entry 2
PS Entry N

Physical Memory

Binghamton University / Qatar University

HPCA 2014
The Permission Store
Augmenting TLBs to Store PS Entries

### TLB

<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>
</tbody>
</table>

...
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 - RW -</td>
</tr>
</tbody>
</table>

TLB
The Memory Permission Manager

Rule

Database

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 Disable
The Memory Permission Manager

Rule
Database

Current Permissions

Requester

New Permissions

Rule
Database

Current Permissions

Requester

New Permissions

Allow or Disallow

Binghamton University / Qatar University

HPCA 2014
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>Hypervisor</td>
<td>R W X</td>
<td>R W X</td>
<td>R W X</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>

None

Wipe Page

Wipe Page

None

None

None
## 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>

---

**Access TLB**

- Hit
- Miss

**Access Pages**

- Read PS
- Entry

**Current Permissions**

**Requester (Current Privilege Level)**

**Access Rule Database**

**Perform Action**

- Write PS
- TLB Match

**Exception**

- No Match
Secure Permission Changes: The PERM_SET Instruction

PERM_SET

%eax, %ebx

Virtual Address

New Permissions

Access TLB
Secure Permission Changes: The PERM_SET Instruction

PERM_SET %eax, %ebx

Virtual Address

Hit

Miss

Access TLB

Access Page Tables

Read PS Entry

Current Permissions

New Permissions

Perf or m Action

Write PS + TLB Match

Exception

No Match

Binghamton University / Qatar University

HPCA 2014

14 / 25
Secure Permission Changes: The PERM_SET Instruction

PERM_SET %eax, %ebx

Access TLB

Miss

Hit

Access Page Tables

Read PS Entry

Current Permissions

Requester (Current Privilege Level)

Virtual Address

New Permissions

PERFOM Action

Write PS + TLB Match

Exception No Match
Secure Permission Changes: The PERM_SET Instruction

PERM_SET

<table>
<thead>
<tr>
<th>Virtual Address</th>
<th>New Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>%eax</td>
<td>%ebx</td>
</tr>
</tbody>
</table>

Access TLB

Hit

Miss

Access Page Tables

Read PS Entry

Current Permissions

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

PERM_SET

Virtual Address
%eax, %ebx

New Permissions

Access TLB

Hit

Miss

Access Page Tables

Read PS Entry

Current Permissions

Requester
(Current Privilege Level)

Access Rule Database
Secure Permission Changes: The PERM_SET Instruction

PERM_SET

\[ \text{PERM}_{-}\text{SET} \ %eax, \ %ebx \]

**Virtual Address**

**New Permissions**

**Access TLB**

**Hit**

**Miss**

**Access Page Tables**

**Current Permissions**

**Requester**

**(Current Privilege Level)**

**Access Rule Database**

**Match**

**Perform Action**

**Write PS + TLB**
Secure Permission Changes: The PERM_SET Instruction

PERM_SET %eax, %ebx

Miss

Access TLB

Hit

Access Page Tables

Read PS Entry

Current Permissions

Requester (Current Privilege Level)

Match

Access Rule Database

No Match

Perform Action

Write PS + TLB

Exception
The Permission Reference Monitor

Load/Store Instructions

<table>
<thead>
<tr>
<th>Expected Permissions</th>
</tr>
</thead>
</table>

Permission Store

<table>
<thead>
<tr>
<th>Actual Permissions</th>
</tr>
</thead>
</table>

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

**Load/Store Instructions**

- Expected Permissions
  - Permission Store

**Permission Store**

- Actual Permissions

Comparison

- Load/Store Instructions
- Permission Store

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

CPU

Core 0

Core 1

PS Entries

ITLB

DTLB

PS Entries

ITLB

DTLB

MMU

Rule Database

PS_Base Register

PERM_SET

Hypervisor

OS

Physical Memory

PS Table

Regular Memory

Protected Memory

Binghamton University / Qatar University

HPCA 2014

16 / 25
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
- Hypervisor
- OS
- Physical Memory
- Regular Memory
- Protected Memory
- PS Table

Binghamton University / Qatar University  HPCA 2014  16 / 25
Hardware Changes Needed for NIMP
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
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

<table>
<thead>
<tr>
<th></th>
<th>IPC Difference (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>calculix</td>
<td>0.2%</td>
</tr>
<tr>
<td>gamess</td>
<td>0.1%</td>
</tr>
<tr>
<td>gromacs</td>
<td>0.3%</td>
</tr>
<tr>
<td>libquantum</td>
<td>3.8%</td>
</tr>
<tr>
<td>mcf</td>
<td>3.5%</td>
</tr>
<tr>
<td>average</td>
<td>1.3%</td>
</tr>
</tbody>
</table>
Reduction in IPC

- calcullix: 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%
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%
- 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 Cache Miss Rate for Regular Accesses

- **Without Perms**
- **With Perms**

Graph showing the miss rate (%) for various benchmarks: calculix, game, gromacs, libquantum, mcf, and average. The graph compares the miss rate with and without permissions, with a noticeable increase in miss rate for mcf when permissions are enabled.
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 (%)

- **Calculix**: Δ 0.3%
- **Games**: Δ 0.08%
- **Gromacs**: Δ 0.4%
- **Libquantum**: Δ 0.4%
- **Mcf**: Δ 0.4%
- **Average**: Δ 0.4%

---

Binghamton University / Qatar University

HPCA 2014
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?