Finding and Exploiting CPU Features using MSR Templating

IEEE Symposium on Security and Privacy 2022

Andreas Kogler
Graz University of Technology

Daniel Weber
CISPA Helmholtz Center for Information Security

Martin Haubenwallner
Graz University of Technology

Moritz Lipp
Amazon Web Services

Daniel Gruss
Graz University of Technology

Michael Schwarz
CISPA Helmholtz Center for Information Security
Agenda

• Motivation
• Framework
  • Detection
  • Classification
  • Extensions
• Case Studies
• **Model Specific Registers (MSRs)**
  - $2^{32}$ 64-bit Registers
  - Documented
  - Undocumented

Motivation
Motivation

- **Model Specific Registers (MSRs)**
  - $2^{32}$ 64-bit Registers
  - Documented
  - Undocumented
- **Influences** on instructions
Motivation

- **Model Specific Registers (MSRs)**
  - $2^{32}$ 64-bit Registers
  - Documented
  - Undocumented
- ** Influences** on instructions
- **Security** patches
Motivation

- **Model Specific Registers** (MSRs)
  - $2^{32}$ 64-bit Registers
  - Documented
  - Undocumented
- **Influences** on instructions
- **Security** patches
- **Hidden** features (e.g., Domas [1])
The Framework: MSR Scanning

MSR Detection

- Scan all MSR addresses
- `rdmsr` → GP-Fault?
- `wrmsr` → GP-Fault?

Complete MSR list

R, W, RW or not-present

Andreas Kogler / twitter @0xhilbert / envelope_alt andreas.kogler@iaik.tugraz.at
The Framework: MSR Scanning

- **Scan all MSR addresses**
  - `rdmsr` → GP-Fault?
  - `wrmsr` → GP-Fault?
The Framework: MSR Scanning

- Scan all MSR addresses
  - rdmsr → GP-Fault?
  - wrmsr → GP-Fault?

✓ Complete MSR list
The Framework: MSR Scanning

- **Scan all** MSR addresses
  - `rdmsr` → GP-Fault?
  - `wrmsr` → GP-Fault?

- Complete MSR list
- \( R, W, RW \) or not-present
The Framework: Documented vs Undocumented

- MSR Detection
  - Parse official PDFs
  - AMD's Reference
  - Intel's SDM
  - Extract table structures
  - Python script
- MSR Scanning
- MSR Classification

Andreas Kogler
@0xhilbert
andreas.kogler@iaik.tugraz.at
The Framework: Documented vs Undocumented

- MSR Detection
  - MSR Scanning
    - Documented
    - Undocumented

- MSR Classification

- Parse official PDFs
  - AMD’s Reference
  - Intel’s SDM
The Framework: Documented vs Undocumented

- **MSR Detection**
  - Documented
  - Undocumented

- **MSR Classification**
  - Parse official PDFs
    - AMD’s Reference
    - Intel’s SDM
  - Extract table structures
    - Python script

Andreas Kogler  
@0xhilbert  
andreas.kogler@iaik.tugraz.at
The Framework: Documented vs Undocumented

**MSR Detection**

- Documented
  - Parse official PDFs
    - AMD’s Reference
    - Intel’s SDM
  - Extract table structures
    - Python script
- Undocumented
  - Documented MSRs
  - Undocumented MSRs
The Framework: Dynamic Analysis

- MSR Detection
- MSR Classification

MSR Scanning

- Documented
- Undocumented

Dynamic Analysis

- Correlation
  - Changing signals
  - Similarity
    - Source
      - Example: MSR 0x637

Similar MSRs

Source hints

Andreas Kogler
@0xhilbert
andreas.kogler@iaik.tugraz.at
The Framework: Dynamic Analysis

- Dynamic MSR:
  - Changing signals
The Framework: Dynamic Analysis

- **MSR Detection**
- **MSR Classification**
- **MSR Scanning**
- Documented
- Undocumented

**Dynamic MSR:**
- Changing signals

**Correlation analysis**
- Similarity
- Source

**Example:** MSR 0x637

Similar MSRs
Source hints

Andreas Kogler
@0xhilbert
andreas.kogler@iaik.tugraz.at
The Framework: Dynamic Analysis

- **Dynamic MSR:**
  - Changing signals
- **Correlation analysis**
  - Similarity
  - Source
- **Example:** MSR 0x637
The Framework: Dynamic Analysis

- Dynamic MSR:
  - Changing signals
- Correlation analysis
  - Similarity
  - Source
- Example: MSR 0x637
  ✓ Similar MSRs
  ✓ Source hints

• Dynamic MSR:
  • Changing signals

• Correlation analysis
  • Similarity
  • Source

• Example: MSR 0x637
  ✓ Similar MSRs
  ✓ Source hints
The Framework: Static Analysis

- MSR Detection
  - MSR Scanning
  - Documented
  - Undocumented

- MSR Classification
  - Dynamic
  - Analyse Correlation

Static MSR:
- Configuration bits
- Execute instruction twice
- Reference
- Modified
- Analyze PMC differences

Influenced instructions
The Framework: Static Analysis

• Static MSR:
  • Configuration bits

Static MSR:
- Configuration bits
- Execute instruction twice
- Reference
- Modified
- Analyze PMC differences
- Influenced instructions

Andreas Kogler
@0xhilbert
andreas.kogler@iaik.tugraz.at
The Framework: Static Analysis

- **Static MSR:**
  - Configuration bits

- **Execute** instruction twice
  - Reference
  - Modified

- Instruction List
- Reference Execution
- Modified Execution
  - PMC differences
  - Influenced instructions
The Framework: Static Analysis

- **Static MSR:**
  - Configuration bits
- **Execute** instruction twice
  - Reference
  - Modified
- **Analyze** PMC differences

![Diagram](image-url)
The Framework: Static Analysis

- **Static MSR:**
  - Configuration bits
- **Execute** instruction twice
  - Reference
  - Modified
- **Analyze** PMC differences
  - Influenced instructions

Instruction List

Reference Execution

Modified Execution

PMCs

Report

Difference?
The Framework: BIOS Templating

- MSR Scanning
  - Documented
  - Undocumented

- MSR Detection
- MSR Classification
  - Dynamic
  - Static

- Analyse Correlation
  - Analyse Bit Effects

- Extend search space
- Change BIOS feature
- Trace differences

- Changed MSRs

Andreas Kogler
@0xhilbert
andreas.kogler@iaik.tugraz.at
The Framework: BIOS Templating

- MSR Detection
- MSR Classification

- Documented
- Dynamic
- Analyse Correlation

- Undocumented
- Static
- Analyse Bit Effects

- Extend search space

BIOS Templating

MSR Scanning & Difference Detection

- Extend search space
- Change BIOS feature
- Trace differences

Changed MSRs

Andreas Kogler / twitter @0xhilbert / envelope_alt andreas.kogler@iaik.tugraz.at
The Framework: BIOS Templating

- MSR Detection
  - Documented
  - Undocumented
  - MSR Scanning
  - Difference Detection
  - Analyse Correlation
  - Analyse Bit Effects

- MSR Classification
  - Dynamic
  - Static

- Extend search space
- Change BIOS feature
The Framework: BIOS Templating

- **MSR Detection**
  - Documented
  - Undocumented

- **MSR Classification**
  - Dynamic
  - Static

- **BIOSTemplating**
  - Extend search space
  - Change BIOS feature
  - Trace differences

Extended search space and trace differences.

Andreas Kogler
@0xhilbert
andreas.kogler@iaik.tugraz.at
The Framework: BIOS Templating

- MSR Detection
  - MSR Scanning & Difference Detection
  - BiOS Templating

- MSR Classification
  - Documented
  - Undocumented
  - Dynamic
  - Static

- Analyse Correlation
  - Analyse Bit Effects

- Extend search space
- Change BIOS feature
- Trace differences

✅ Changed MSRs

Andreas Kogler
@0xhilbert
andreas.kogler@iaik.tugraz.at
The Framework: Summary

- BIOS Templating
- MSR Scanning & Difference Detection
- Documented
- Dynamic
- Analyse Correlation
- Undocumented
- Static
- Analyse Bit Effects

List (R, W, RW, or NP)
- Dynamic: similar MSRs
- Static: influenced instruction
- BIOS: changed MSRs
The Framework: Summary

- **BIOS Templating**
- **MSR Scanning & Difference Detection**
- **Documented**
- **Undocumented**
- **Dynamic**
- **Static**
- **Analyse Correlation**
- **Analyse Bit Effects**

✓ List \( R, W, RW, \) or \( NP \)
The Framework: Summary

- MSR Detection
  - Documented
  - Undocumented
- MSR Classification
  - Dynamic
  - Static
- Analyse Correlation
- Analyse Bit Effects

- List (R, W, RW, or NP)
- Dynamic: similar MSRs

BIOS Templating
MSR Scanning & Difference Detection
The Framework: Summary

- MSR Detection
  - Documented
  - Undocumented
- MSR Classification
  - Dynamic
  - Static
- Analyse Correlation
- Analyse Bit Effects

- List (R, W, RW, or NP)
- Dynamic: similar MSRs
- Static: influenced instruction
The Framework: Summary

- **MSR Detection**
  - Documented
  - Undocumented

- **MSR Classification**
  - Dynamic
  - Static

- **BIOS Templating & MSR Scanning & Difference Detection**

- List (R, W, RW, or NP)
- Dynamic: similar MSRs
- Static: influenced instruction
- BIOS: changed MSRs

---

@AndreasKogler @0xhilbert andreas.kogler@iaik.tugraz.at
Case Studies

- Attack case studies

AES-NI
Xen
uCode
Case Studies

- **Attack** case studies
- **Defense** case studies

- AES-NI
- Prefetch
- CrossTalk
- Medusa
- Xen
- uCode

Andreas Kogler / twitter @0xhilbert / envelope_alt andreas.kogler@iaik.tugraz.at
Case Study: Prefetch

- Prefetch-based attacks [2]

[Graph showing kernel offset in MB vs. cycles]
Case Study: Prefetch

- Prefetch-based attacks [2]

---

Kernel offset in MB

Cycles

65 70 75 80 85

60 80 100 120 140

Andreas Kogler / twitter: @0xhilbert / envelope_alt andreas.kogler@iaik.tugraz.at
## Case Study: Prefetch

<table>
<thead>
<tr>
<th>Instruction</th>
<th>MSR</th>
<th>PMC Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREFETCHNTA</td>
<td>Bit 2</td>
<td>-1 LdDispatch</td>
</tr>
<tr>
<td>PREFETCHT0</td>
<td>Bit 3</td>
<td>-1 LdDispatch</td>
</tr>
<tr>
<td>PREFETCHT1</td>
<td>Bit 4</td>
<td>-1 LdDispatch</td>
</tr>
<tr>
<td>PREFETCHT2</td>
<td>Bit 5</td>
<td>-1 LdDispatch</td>
</tr>
<tr>
<td>PREFETCHW</td>
<td>Bit 6</td>
<td>-1 LdDispatch</td>
</tr>
<tr>
<td>PREFETCH</td>
<td>Bit 7</td>
<td>-1 LdDispatch</td>
</tr>
</tbody>
</table>

- **Prefabch-based** attacks [2]
- **Search** configuration bits
# Case Study: Prefetch

<table>
<thead>
<tr>
<th>Instruction</th>
<th>MSR</th>
<th>PMC Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREFETCHNTA</td>
<td>Bit 2</td>
<td>-1 LdDispatch</td>
</tr>
<tr>
<td>PREFETCHT0</td>
<td>Bit 3</td>
<td>-1 LdDispatch</td>
</tr>
<tr>
<td>PREFETCHT1</td>
<td>Bit 4</td>
<td>-1 LdDispatch</td>
</tr>
<tr>
<td>PREFETCHT2</td>
<td>Bit 5</td>
<td>-1 LdDispatch</td>
</tr>
<tr>
<td>PREFETCHW</td>
<td>Bit 6</td>
<td>-1 LdDispatch</td>
</tr>
<tr>
<td>PREFETCH</td>
<td>Bit 7</td>
<td>-1 LdDispatch</td>
</tr>
</tbody>
</table>

- **Prefetch-based** attacks [2]
- Search configuration bits
- **Disable** prefetch

---

Andreas Kogler
Andreas Kogler
andreas.kogler@iaik.tugraz.at
Case Study: Prefetch

- Prefetch-based attacks [2]
- Search configuration bits
- Disable prefetch*
  ✓ No prefetch-based attacks
Case Study: Prefetch

- **Prefetch-based** attacks [2]
- **Search** configuration bits
- **Disable** prefetch*
- ✔ **No** prefetch-based attacks
- ✔ **1%** Binaries $\rightarrow$ **0.04%** SPEC

---

*Disable prefetch is not explicitly mentioned in the image but is inferred from the context.
Case Study: AES-NI

- Lock bit
Case Study: AES-NI

- Lock bit
- Disable at runtime

Full key

Andreas Kogler / twitter @0xhilbert / envelope_alt andreas.kogler@iaik.tugraz.at
Case Study: AES-NI

```c
/* ... */
if( mbedtls_aesni_has_support( MBEDTLS_AESNI_AES ) )
    return( mbedtls_aesni_setkey_enc( ctx->rk, key, keybits ) );
/* ... */
switch( ctx->nr ) {
    case 10:
        for( i = 0; i < 10; i++, RK += 4 ) {
            RK[4] = RK[0] ^ RCON[i] ^
                ( FSb[ ( RK[3] >> 8 ) & 0xFF ] ) ^
                ( FSb[ ( RK[3] >> 16 ) & 0xFF ] << 8 ) ^
                ( FSb[ ( RK[3] >> 24 ) & 0xFF ] << 16 ) ^
                ( FSb[ ( RK[3] ) & 0xFF ] << 24 );
        }
        break;
/* additional cases for different key lengths */
}
/* ... */
```

- Lock bit
- Disable at runtime
- MbedTLS in SGX
if( mbedtls_aesni_has_support( MBEDTLS_AESNI_AES ) )
    return( mbedtls_aesni_setkey_enc( ctx->rk, key, keybits ) );

switch( ctx->nr ) {
    case 10:
        for( i = 0; i < 10; i++, RK += 4 ) {
            RK[4] = RK[0] ^ RCON[i] ^
            ( FSb[ ( RK[3] >> 8 ) & 0xFF ] ) ^
            ( FSb[ ( RK[3] >> 16 ) & 0xFF ] << 8 ) ^
            ( FSb[ ( RK[3] >> 24 ) & 0xFF ] << 16 ) ^
            ( FSb[ ( RK[3] ) & 0xFF ] << 24 );
        }
        break;
    /* additional cases for different key lengths */
}

• Lock bit
• Disable at runtime
• MbedTLS in SGX
  • Check AES-NI
Case Study: AES-NI

```c
/* ... */
if( mbedtls_aesni_has_support( MBEDTLS_AESNI_AES ) )
    return( mbedtls_aesni_setkey_enc( ctx->rk, key, keybits ) );
/* ... */
switch( ctx->nr ) {
    case 10:
        for( i = 0; i < 10; i++, RK += 4 ) {
            RK[4] = RK[0] ^ RCON[i] ^
                ( FSb[ ( RK[3] >> 8 ) & 0xFF ] ) ^
                ( FSb[ ( RK[3] >> 16 ) & 0xFF ] << 8 ) ^
                ( FSb[ ( RK[3] >> 24 ) & 0xFF ] << 16 ) ^
                ( FSb[ ( RK[3] ) & 0xFF ] << 24 );
        }
        break;
    /* additional cases for different key lengths */
}
/* ... */
```

- **Lock bit**
- **Disable at runtime**
- **MbedTLS in SGX**
  - Check AES-NI
  - Fallback T-Tables
Case Study: AES-NI

- Lock bit
- Disable at runtime
- MbedTLS in SGX
  - Check AES-NI
  - Fallback T-Tables
  - LLC P+P
Case Study: AES-NI

- Lock bit
- Disable at runtime
- MbedTLS in SGX
  - Check AES-NI
  - Fallback T-Tables
  - LLC P+P
Case Study: AES-NI

- Lock bit
- Disable at runtime
- MbedTLS in SGX
  - Check AES-NI
  - Fallback T-Tables
  - LLC P+P
  - Z3 Solver

![Graph showing Memory Accesses vs Truncated Set Index with data points for $K_1$ and $K_2$.]
Case Study: AES-NI

- Lock bit
- Disable at runtime
- MbedTLS in SGX
  - Check AES-NI
  - Fallback T-Tables
  - LLC P+P
  - Z3 Solver

✓ Full key
Case Study: CrossTalk

- CrossTalk attack [3]

Andreas Kogler / twitter @0xhilbert / envelope_alt andreas.kogler@iaik.tugraz.at
Case Study: CrossTalk

- **CrossTalk** attack [3]
- **Unpriviledged** leakage
  - cpuid $\rightarrow 88.9\%$
  - rdseed $\rightarrow 0.4\%$
Case Study: CrossTalk

- CrossTalk attack [3]
- Unprivileged leakage
  - cpuid $\rightarrow$ 88.9%
  - rdseed $\rightarrow$ 0.4%
- Search configuration bits
Case Study: CrossTalk

- CrossTalk attack [3]
- Unprivileged leakage
  - cpuid $\rightarrow 88.9\%$
  - rdseed $\rightarrow 0.4\%$
- Search configuration bits
- CPUID trap
Case Study: CrossTalk

- CrossTalk attack [3]
- Unprivileged leakage
  - cpuid \rightarrow 88.9\%
  - rdseed \rightarrow 0.4\%
- Search configuration bits
- CPUID trap

✓ Reduced by 211.4 times
Case Study: Xen Foreshadow

Hardware:  
Xen HV:  
Guest:  
\[ \text{rdmsr} \]

- Hypervisor handles MSRs

• Hypervisor handles MSRs
Case Study: Xen Foreshadow

- **Hypervisor** handles MSRs
- **XEN** deny list

Hardware:

Xen HV:

- rdmsr

Guest:
Case Study: Xen Foreshadow

Hardware:

Xen HV:
- rdmsr
- check

Guest:
- hypervisor handles MSRs
- XEN deny list

Leak 214 Byte/s
Andreas Kogler / twitter @0xhilbert / envelope_alt andreas.kogler@iaik.tugraz.at
Case Study: Xen Foreshadow

Hardware:
- MSR 0x637

Xen HV:
- check
- rdmsr

Guest:
- Hypervisor handles MSRs
- XEN deny list
- Unrestricted read access

Andreas Kogler /twitter @0xhilbert /envelope_alt andreas.kogler@iaik.tugraz.at
Case Study: Xen Foreshadow

Hardware:
- MSR 0x637

Xen HV:
- check
- forward
- rdmsr

Guest:

- Hypervisor handles MSRs
- XEN deny list
- Unrestricted read access
- Timer MSR

Andreas Kogler / twitter @0xhilbert / envelope_alt andreas.kogler@iaik.tugraz.at
Case Study: Xen Foreshadow

- Hypervisor handles MSRs
- XEN deny list
- Unrestricted read access
- Timer MSR
  - Cache hit vs miss
Case Study: Xen Foreshadow

- Hypervisor handles MSRs
- XEN deny list
- Unrestricted read access
- Timer MSR
  - Cache hit vs miss
  - Foreshadow attack [4]
Case Study: Xen Foreshadow

- Hypervisor handles MSRs
- XEN deny list
- Unrestricted read access
- Timer MSR
  - Cache hit vs miss
  - Foreshadow attack [4]

✓ Leak 214 Byte/s
Case Study: uCode Diffing

- Analyze \( \mu \)-Code Patches

\( \mu \)-Code Patches

Reference Execution

Patched Execution

Difference Detection

Report
Case Study: uCode Diffing

- Analyze $\mu$-Code Patches
- Detect new

$\mu$-Code Patches

Reference Execution

Patched Execution

Difference Detection

Report

Andreas Kogler
@0xhilbert
andreas.kogler@iaik.tugraz.at
Case Study: uCode Diffing

- Analyze $\mu$-Code Patches
- Detect new
- Detect affected instructions

$\mu$-Code Patches

Reference Execution

Patched Execution

Difference Detection

Report
Case Study: µCode Diffing

- Analyze µ-Code Patches
- Detect new
- Detect affected instructions
✓ Before public disclosure
Conclusion

- **Framework** https://github.com/IAIK/msrevelio
Conclusion

• Framework [GitHub](https://github.com/IAIK/msrevelio)
• Case Studies

For more details ...
Conclusion

• **Framework** [GitHub](https://github.com/IAIK/msrevelio)

• **Case Studies**

• **MSRs** enable defenses
Conclusion

- **Framework** [GitHub](https://github.com/IAIK/msrevelio)
- **Case Studies**
- **MSRs** enable defenses
- **MSRs** open new attack vectors
Conclusion

- Framework ☑ https://github.com/IAIK/msrevelio
- Case Studies
- MSRs enable defenses
- MSRs open new attack vectors
- For more details ...

Andreas Kogler  @0xhilbert  andreas.kogler@iaik.tugraz.at
Conclusion

- **Framework** [GitHub](https://github.com/IAIK/msrevelio)
- **Case Studies**
- **MSRs** enable defenses
- **MSRs** open new attack vectors
- For more details, read the paper.

Andreas Kogler  
@0xhilbert  
andreas.kogler@iaik.tugraz.at


# Overall Results

<table>
<thead>
<tr>
<th>CPU</th>
<th>AMD</th>
<th>Intel</th>
<th>Intel</th>
<th>Intel</th>
<th>Intel</th>
<th>Intel</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Threadripper 1920X</td>
<td>i7-6700k</td>
<td>i7-8700k</td>
<td>i9-9900k</td>
<td>Xeon Silver 4208</td>
<td></td>
</tr>
<tr>
<td><strong>µ-Arch</strong></td>
<td>Zen</td>
<td>Skylake</td>
<td>Coffee Lake</td>
<td>Coffee Lake</td>
<td>Cascade Lake</td>
<td></td>
</tr>
<tr>
<td><strong>µ-Code</strong></td>
<td>0x8001137</td>
<td>0x9e</td>
<td>0xb4</td>
<td>0xde</td>
<td>0x5003102</td>
<td></td>
</tr>
<tr>
<td># Found(^1)</td>
<td>5244 (5223, 17, 4)</td>
<td>477 (363, 108, 5)</td>
<td>517 (388, 122, 7)</td>
<td>537 (413, 117, 7)</td>
<td>1109 (957, 142, 10)</td>
<td></td>
</tr>
<tr>
<td># Undoc(^1)</td>
<td>4876 (4873, 2, 1)</td>
<td>105 (68, 35, 2)</td>
<td>126 (89, 35, 2)</td>
<td>136 (99, 35, 2)</td>
<td>647 (591, 52, 4)</td>
<td></td>
</tr>
<tr>
<td># Static(^2)</td>
<td>4873 (4871, 2)</td>
<td>99 (68, 31)</td>
<td>121 (89, 32)</td>
<td>132 (99, 33)</td>
<td>601 (553, 48)</td>
<td></td>
</tr>
<tr>
<td># Dynamic(^2)</td>
<td>2 (2, 0)</td>
<td>4 (0, 4)</td>
<td>3 (0, 3)</td>
<td>2 (0, 2)</td>
<td>42 (38, 4)</td>
<td></td>
</tr>
<tr>
<td># Similar</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>42</td>
<td></td>
</tr>
</tbody>
</table>

\(^1\sum (RW, RO, WO)\) \(^2\sum (RW, RO)\)