Whispers in the Hyper-space: High-speed Covert Channel Attacks in the Cloud

Zhenyu Wu†, Zhang Xu, Haining Wang

The College of William and Mary

† Now affiliated with NEC Laboratories America Inc.

The College of WILLIAM & MARY

Virtualization and Cloud Computing

- Server Virtualization
  - Consolidates workload
  - Simplifies resource management
  - Enabling Utility-based Cloud Computing
- Server Virtualization Technologies
  - Goal: Computing consolidation
  - Design: Logically-separate but physically-shared
  - NOT equivalent to physically separated machines
    - Non-negligible differences expected, e.g. covert channels

The College of WILLIAM & MARY

The Threat of Covert Channels

- Covert Channels
  - Exploit imperfection in the isolation of shared resources
  - Enable communication bypassing mandatory access controls (i.e. data exfiltration)
- On non-virtualized systems
  - File system objects, network stack, shared processor cache, input device, etc.
- On virtualized systems
  - No practical exploit to date

The College of WILLIAM & MARY

Related Work

- Precautionary Research
  - Hey, you, get off of my cloud (Ristenpart et al., CCS'09)
    - Bandwidth 0.02bps << 1bps by TCSEC
  - Follow up research (Xu et al., CCSW'11)
    - Improved bandwidth 3.2bps (~10% error rate)
    - Conclusion: can do very limited harm in the cloud
- Industry Response
  - Amazon EC2 Dedicated Instances
    - Significant extra-cost (Targeting the "paranoid high-end"?)

The College of WILLIAM & MARY

Covert Channel Attacks are REAL and PRACTICAL!

The College of William & Mary

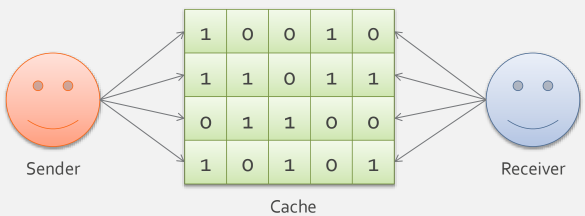
Project Outline

- Understand Existing Cache Covert Channel
  - Why it doesn't work in the Cloud
- Design a New Covert Channel
  - Targeting cross-VM communication
- Realistic Evaluation
  - In-house and on Amazon EC2

The College of William & Mary

Struggles of Cache Covert Channel

- Classic Cache Channel
  - A hybrid timing & storage channel



The College of William & Mary

Struggles of Cache Covert Channel

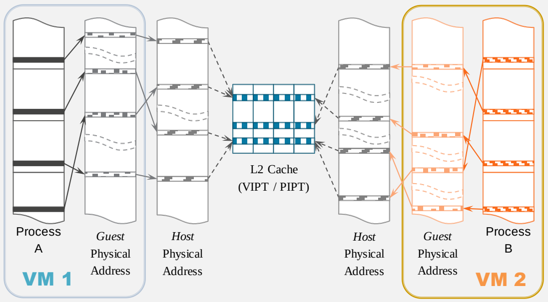
- Classic Cache Channels
  - Works very well on hyper-threaded processor
    - Bandwidth ~400 kilo bytes/sec

      ~3,000,000 bits/sec
  - Performs very poorly on virtualize environment
    - Bandwidth ~1 bit/sec
- Reasons for Struggling
  - Addressing Uncertainty
  - Scheduling Uncertainty
  - Cache Physical Limitation

The College of William & Mary

Struggle #1: Addressing Uncertainty

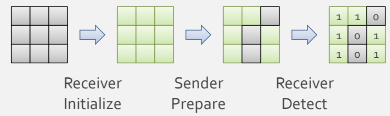
- Virtualization
  - Additional Layer of Indirection (or, complication)



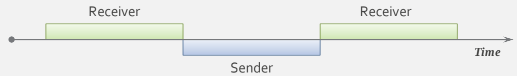
The College of William & Mary

Struggle #2: Scheduling Uncertainty

- Cache "Round-trip" Requirement



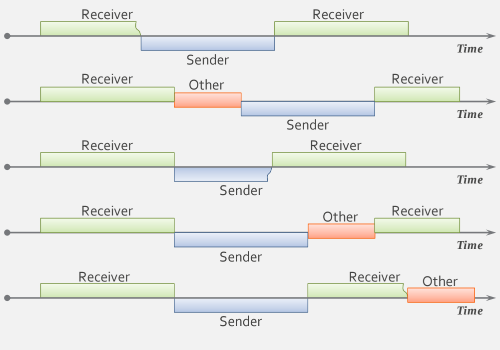
- Strict Round-Robin Scheduling



The College of William & Mary

Struggle #2: Scheduling Uncertainty

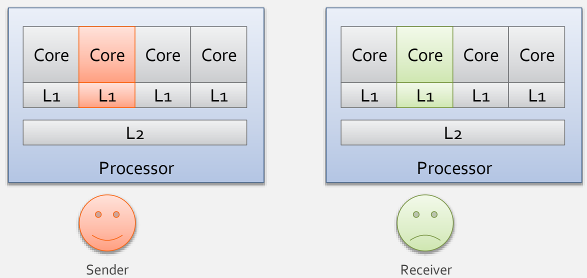
- Too Many Invalid Scheduling Alternatives



The College of William & Mary

Struggle #3: Physical Limitation

- Require stably shared cache, however
  - Non-participating workload destabilize the cache
  - Cores between physical processors do not share cache!



The College of William & Mary

Project Outline

- Understand Existing Cache Covert Channel
  - Why it doesn't work in the Cloud
- Design a New Covert Channel
  - Targeting cross-VM communication
- Realistic Evaluation
  - In-house and on Amazon EC2

The College of William & Mary

Redesigning Data Transmission

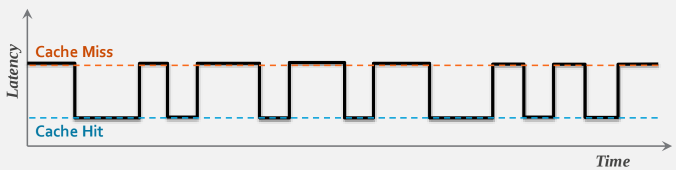
- Change of Data Modulation Technique
  - Cache region based encoding no longer suitable
  - State Patterns ➔ Timing Patterns

1	0	0	1	0
1	1	0	1	1
0	1	1	0	0
1	0	1	0	1

The College of William & Mary

Redesigning Data Transmission

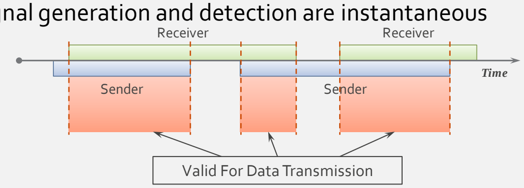
- Change of Data Modulation Technique
  - Cache region based encoding no longer suitable
  - State Patterns ➔ Timing Patterns



The College of William & Mary

Redesigning Data Transmission

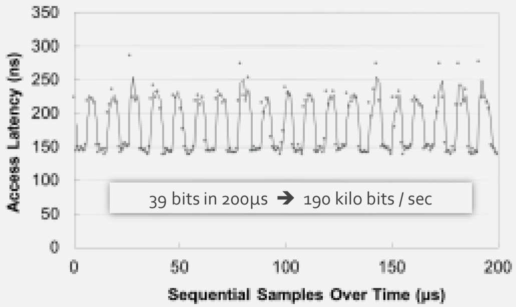
- Change of Data Modulation Technique
  - Cache region based encoding no longer suitable
  - State Patterns ➔ Timing Patterns
- Change of Scheduling Requirement
  - Signal generation and detection are instantaneous



The College of William & Mary

Demonstration

- L2 cache channel using new transmission scheme
  - Intel Core2, Hyper-V, Windows Guests



The College of William & Mary

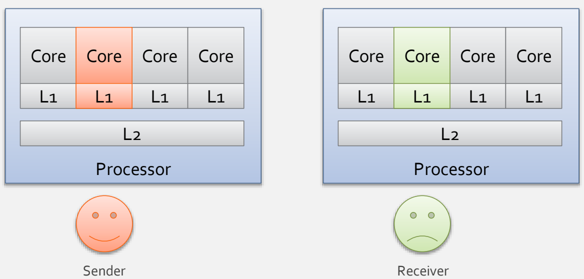
Progress So Far…

- Addressing Uncertainty
- Scheduling Uncertainty
- Cache Physical Limitation

The College of William & Mary

An Insurmountable Limitation

- A shared cache is not guaranteed
  - Virtual processor migration results in frequent interruption



The College of William & Mary

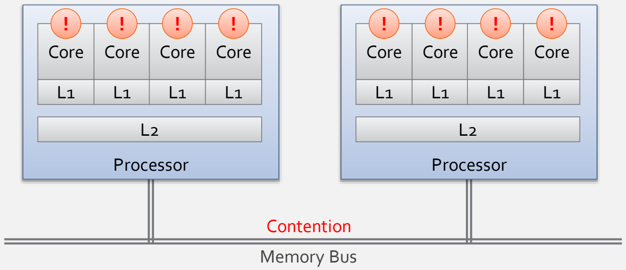
The Memory Bus: Fibers of the Hyper-space

The College of William & Mary

The Bus Contention Channel

- Look beyond the processor cache…

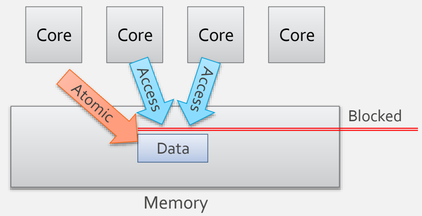
  … we see the memory bus



Exploiting Atomic Instructions

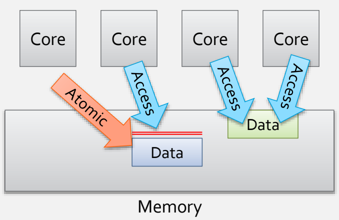
- Atomic Instructions
  - Special x86 memory manipulations
    - Ensures operation atomicity
  - Very useful for thread synchronization…

    … and covert channels



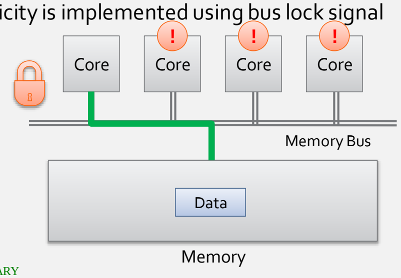
Exploiting Atomic Instructions

- Theory vs. Reality
  - Theoretical
    - Performance impact is localized



Exploiting Atomic Instructions

- Theory vs. Reality
  - Theoretical (Not Exploitable)
    - Performance impact is localized
  - Pre-Pentium Pro
    - Atomicity is implemented using bus lock signal



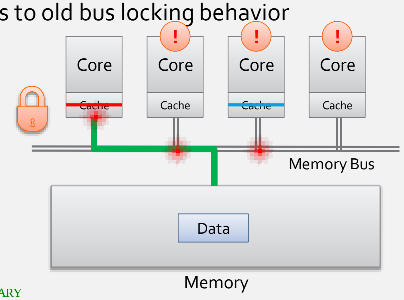
Exploiting Atomic Instructions

- Theory vs. Reality
  - Theoretical (Not Exploitable)
    - Performance impact is localized
  - Pre-Pentium Pro (Exploitable)
    - Atomicity is implemented using bus lock signal
  - Pentium Pro ~ Pre-Nehalem
    - Bus lock is reduced, but not eliminated

The College of William & Mary

Exploiting Atomic Instructions

- When target data can be put on a cache line
  - Cache line locking is performed
- But when target data spans two cache lines…
  - Reverts to old bus locking behavior



The College of William & Mary

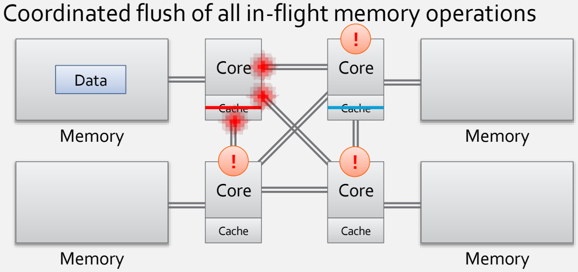
Exploiting Atomic Instructions

- Theory vs. Reality
  - Theoretical (Not Exploitable)
    - Performance impact is localized
  - Pre-Pentium Pro (Exploitable)
    - Atomicity is implemented using bus lock signal
  - Pentium Pro ~ Pre-Nehalem (Exploitable)
    - Bus lock is reduced, but not eliminated
  - Nehalem ~ ?
    - No shared memory bus!

The College of William & Mary

Exploiting Atomic Instructions

- When target data can be put on a cache line
  - Cache line locking is performed
- But when target data spans two cache lines…
  - Coordinated flush of all in-flight memory operations



The College of William & Mary

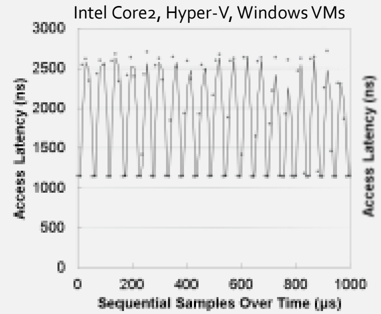
Exploiting Atomic Instructions

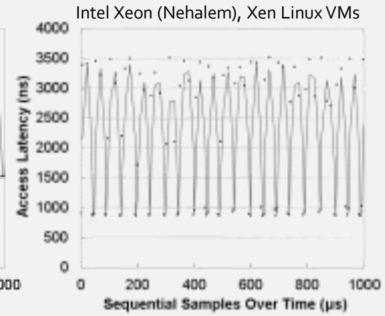
- Theory vs. Reality
  - Theoretical (Not Exploitable)
    - Performance impact is localized
  - Pre-Pentium Pro (Exploitable)
    - Atomicity is implemented using bus lock signal
  - Pentium Pro ~ Pre-Nehalem (Exploitable)
    - Bus lock is reduced, but not eliminated
  - Nehalem ~ ? (Still Exploitable)
    - No shared memory bus!

The College of William & Mary

Demonstration

- Memory Bus Channels





39 bits in 1ms → 38 kilo bits / sec

The College of William & Mary

So Far, So Good… And Then?

- Addressing Uncertainty ✔
- Scheduling Uncertainty ✔
- Cache Physical Limitation ✔

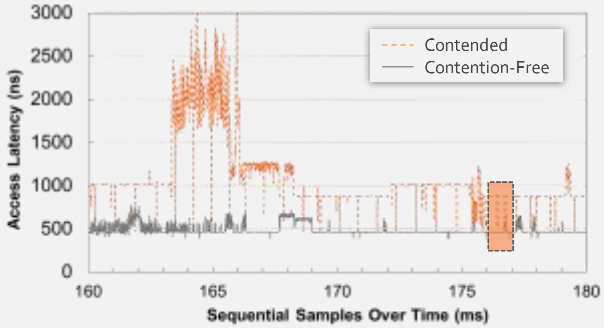
The College of William & Mary

Space Communication is Hard!

The College of William & Mary

Reliable Communication

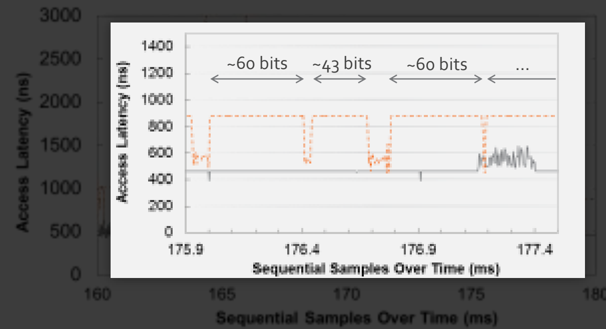
- The memory bus channel is quite noisy



The College of William & Mary

Reliable Communication

- The memory bus channel is quite noisy



The College of William & Mary

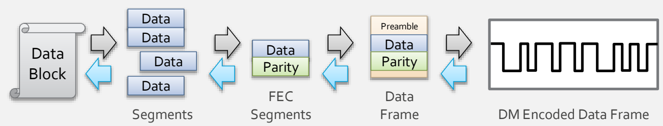
Reliable Communication

- Receiving Confirmation
  - No Explicit Send-and-Ack
  - Simultaneous receiver detection
    - Leveraging the system-wide observable bus contention
- Clock Synchronization
  - Self-clock coding – Differential Manchester Coding
    - Standard network coding used in token ring network
- Error Correction
  - Again, no Send-and-Ack
  - Forward Error Correction – Reed-Solomon Coding
    - Use for optical discs, satellite communication, etc.

The College of William & Mary

Reliable Communication

- Data Framing
  - Send data in small fixed-length packets
  - Improves resilience to interruptions and errors
- Overview of Complete Protocol



The College of William & Mary

Project Outline

- Understand Existing Cache Covert Channel
  - Why it doesn't work in the Cloud
- Design a New Covert Channel
  - Targeting cross-VM communication
- Realistic Evaluation
  - In-house and on Amazon EC2

The College of William & Mary

In-house Experiment

- Realistic Setup
  - Intel Xeon (Nehalem), Xen, Linux Guests
  - Spawn two VMs, single vCPU, 512MB memory
  - Sender and receiver runs as unprivileged user programs
- Protocol Parameters
  - 32 bit data frame (8 bit preamble, 24 bit payload)
  - FEC Parity : Data = 4 : 4
- Channel Capacity
  - Bandwidth: 781.6 $\pm$ 13.6 bps
  - Error rate: 0.27%

The College of William & Mary

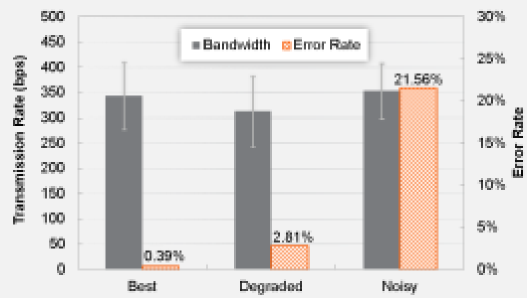
Amazon EC2 Experiments

- Preparation
  - Spawn two physical co-resident VMs
    - /proc/cpuinfo: Intel Xeon (Yorkfield) (share memory bus)
- Initial Trial:
  - Protocol Parameters
    - 24 bit data frame (8 bit preamble, 16 bit payload)
    - FEC Parity : Data = 4 : 4
  - Best Channel Capacity
    - Bandwidth: 343.5 $\pm$ 66.1 bps
    - Error rate: 0.39%
    - But degrades over time in an interesting way

The College of William & Mary

Amazon EC2 Experiments

- Three Staged Performance
  - Due to Xen scheduler preemption



The College of William & Mary

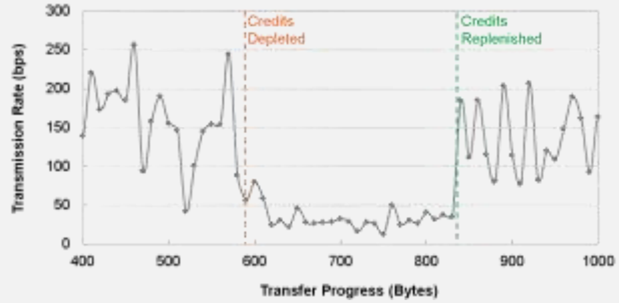
Amazon EC2 Experiments

- Adapting to "Offensive" scheduling:
  - Protocol Parameters
    - Slow down transmission rate by 20%
    - Increase FEC Parity : Data = 4 : 2
  - Stable Channel Capacity
    - Bandwidth: 107.9 ± 39.9 bps
    - Error rate: 0.75%

The College of WILLIAM & MARY

Amazon EC2 Experiments

- Protocol Versatility
  - Automatic speed adjustment according to channel quality



The College of WILLIAM & MARY

How to Mitigate the Problem?

The College of WILLIAM & MARY

Mitigation Avenues

- Tenants
  - Limited Options – Performance Anomaly Dentations
    - Defender continuously monitors memory access latency
    - High overhead, high false positives
- Cloud Providers
  - Preventative Measures
    - Dedicated Instance (Amazon EC2) – Effective but costly
    - Alternative Low-Cost Placement Strategies
      - Controlled and deterministic sharing
      - Makes arbitrary attack difficult

The College of WILLIAM & MARY

Mitigation Avenues

- Cloud Providers
  - Detective Measures
    - Leverage access to Hypervisor, and rich server resource
    - Discover performance anomalies by hardware counters
    - Low overhead, low false positive penalties
- Hardware Manufactures
  - Isolation Improvements (Preventative Measures)
    - Trap instead of handling exotic conditions
    - Tag request by VM, and limit performance impact scope
  - Costly (one-time), but effective and efficient

The College of WILLIAM & MARY

Conclusion

- Covert Channel Attacks in the Cloud are Practical
  - We uncovered a high speed covert channel medium
  - We designed a reliable transmission protocol
  - We evaluated the performance in a real cloud environment
- Call for Effective and Efficient Mitigations
  - Three avenues of approaches:
    - Tenant
    - Cloud Provider
    - Hardware Manufacture

The College of WILLIAM & MARY

Q & A

Thank you for listening!

The College of WILLIAM & MARY