NC STATE UNIVERSITY Computer Science

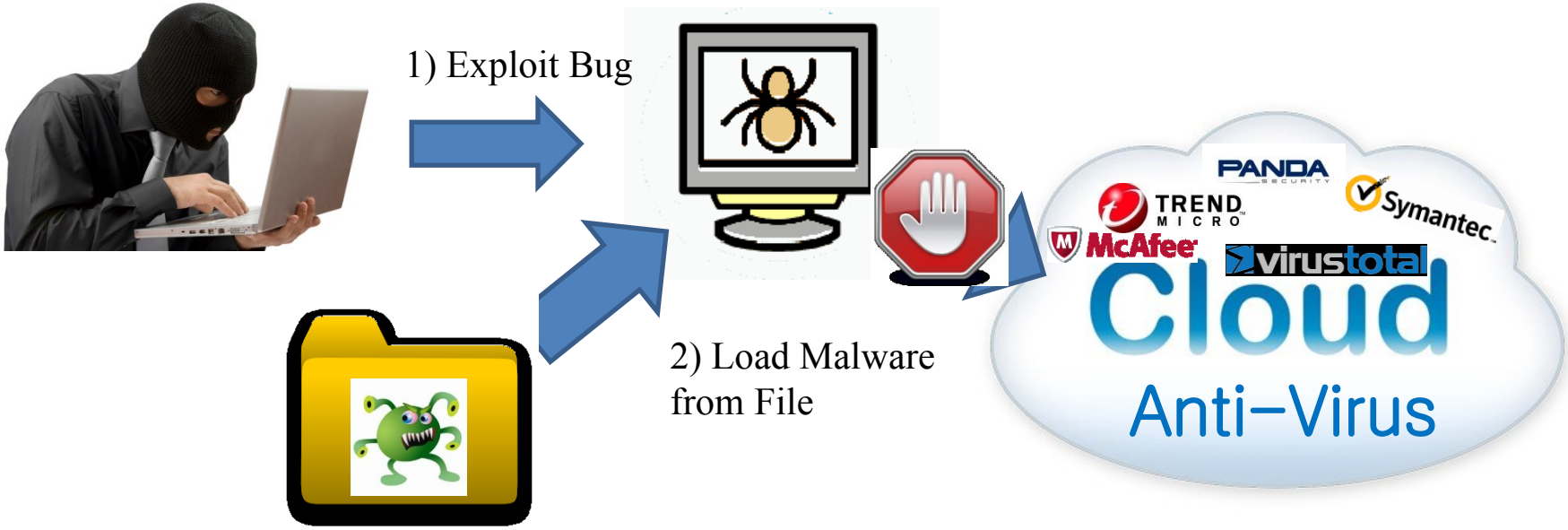
SEER: Practical Memory Virus Scanning-as-a-Service for Virtualized Environments

Jason Gionta, Ahmed Azab,
William Enck, Peng Ning,
Xiaolan Zhang

Data Centers and Security Software

Offload Analysis to Cloud

- System Compromise



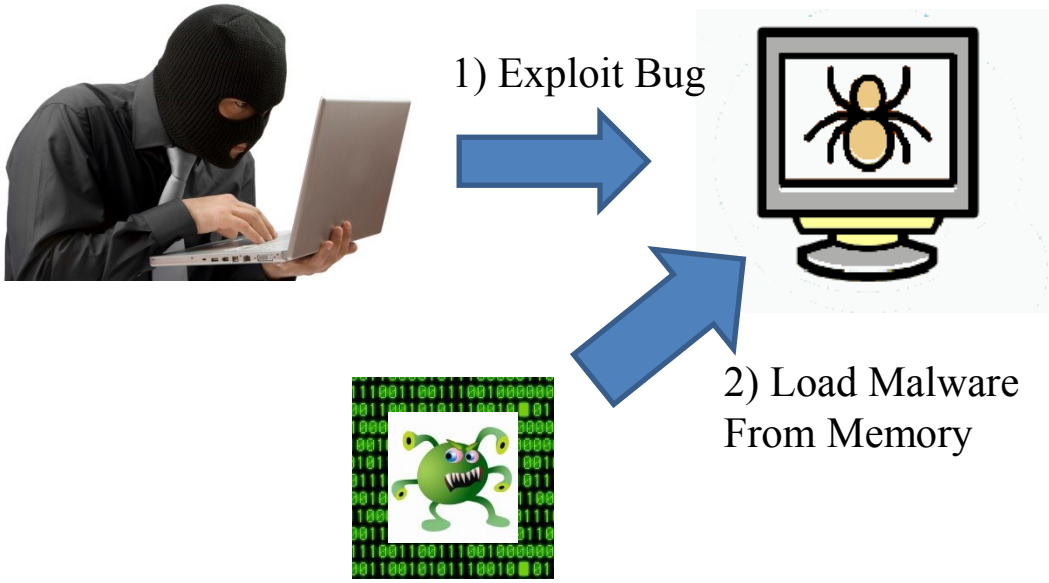
- Cloud Based Virus Scanning – Virus Scanning-as-a-Service (VSaaS)
  - Hook file operations
    - Check against all known files
    - Scan file otherwise

Virus Scanning-as-a-Service (VSaaS)

- Products and Research
  - Logical VHD Scanning [Wei et al. 2009]
    - Scanning of VHD's offline
  - Scan-Lite [Soules et al. 2009]
    - Enterprise file scanner scheduling
  - CloudAV [Oberhiede et al. 2008]
    - Network Service for scanning files on demand
  - VirusTotal
  - Network Scanning Appliances
    - TrendMicro "Deep Security"
    - Symantec End-Point Security Appliance
- Does not support memory scanning

Memory-Only Malware

- System Compromise

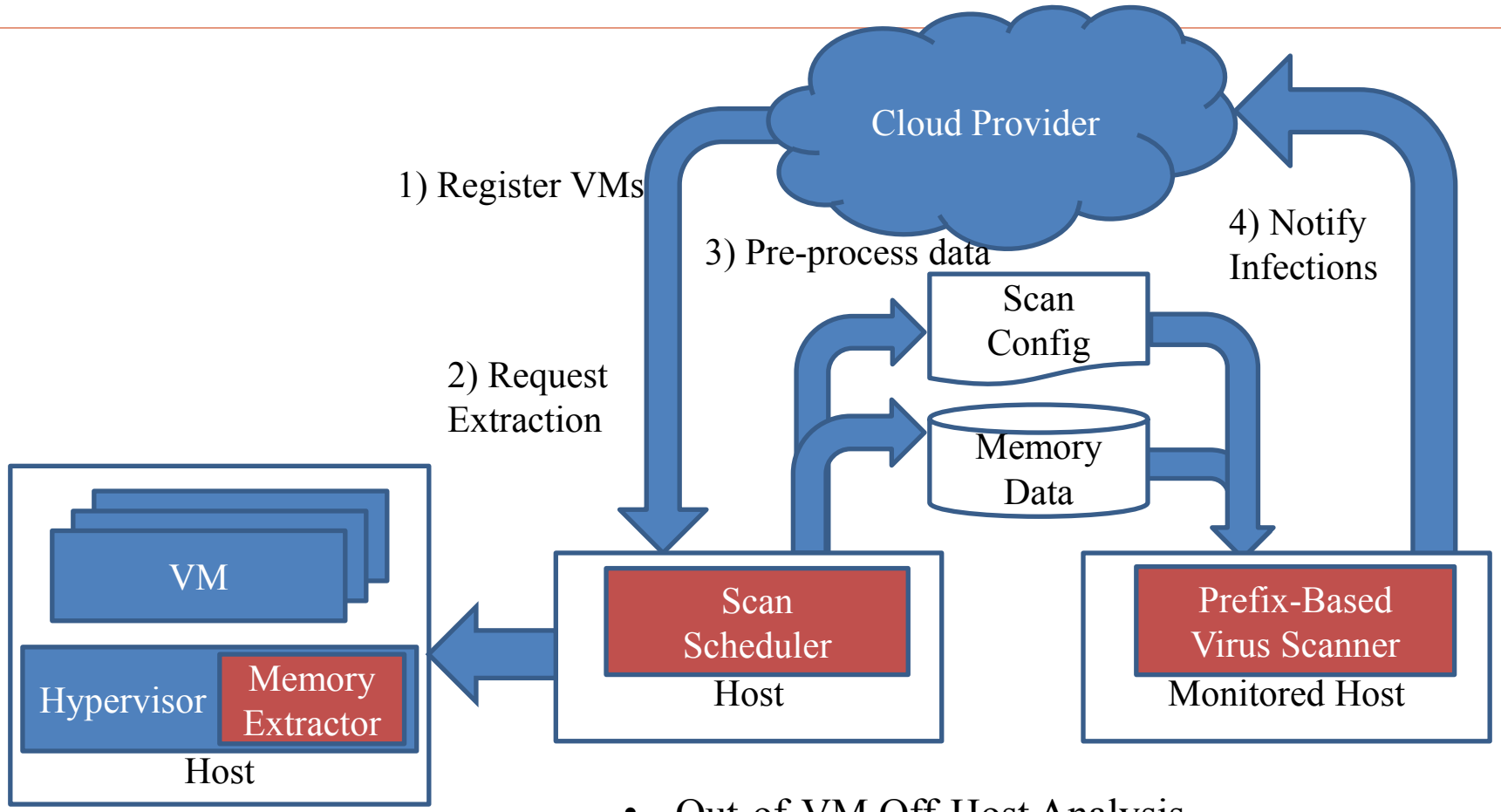


- Trades persistence for stealth
- Bypasses file-based Virus Scanners

Challenges: Scanning Memory

- Must **scan all memory** for compromise
  - Virus Scanning is computationally intensive
- **Changing memory can impact correctness**
- **Resource contention**
  - Anti-Virus Storms
- **File based optimizations not suitable for memory**
  - File hashing

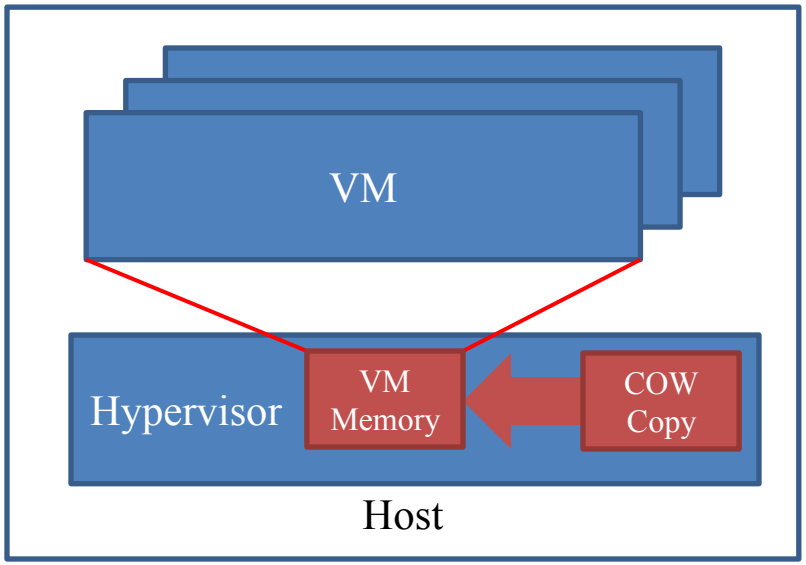
SEER Architecture & Features



- Out-of-VM Off-Host Analysis
- Deduplicated Data Shipping
- Pre-Process Data
- Reduce Cost of Scanning

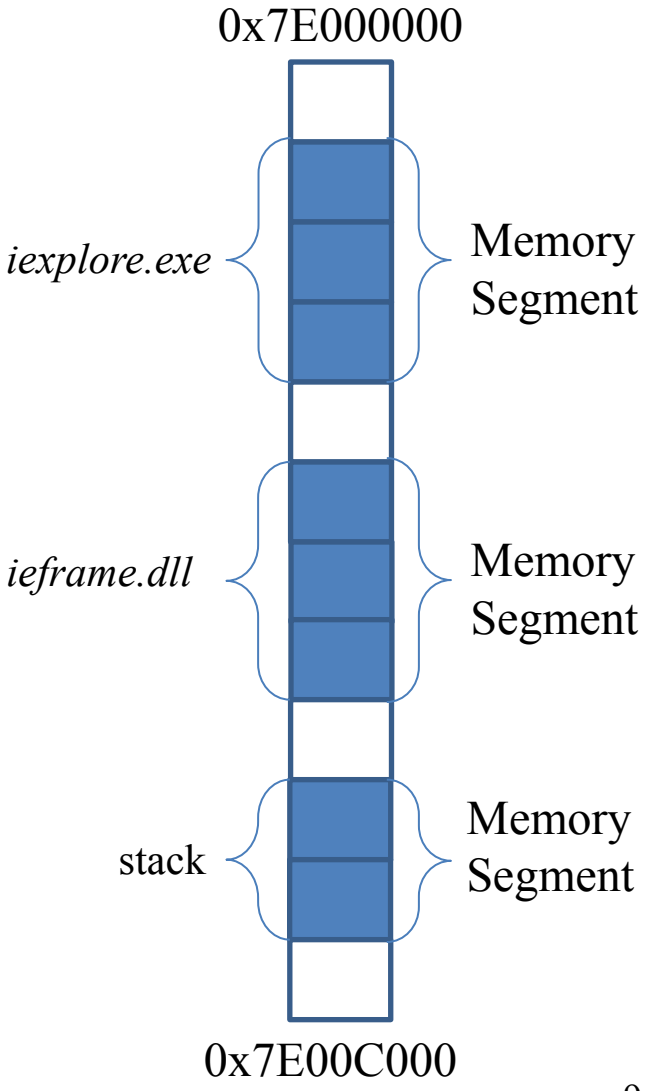
Out-of-VM Off-Host Analysis

- Fast VM Snapshots
  - Create copy of running VM memory
    - Copy-On-Write



Out-of-VM Off-Host Analysis

- *Memory Carving*
  - Extract memory into *Memory Segments**
    - OS managed memory allocations
      - e.g., heaps, stacks, memory mapped files



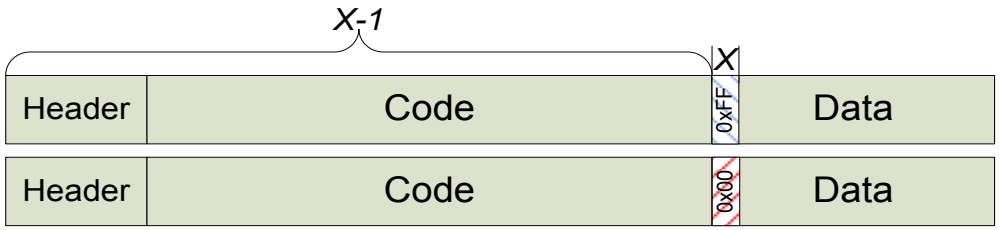
* Not x86 Segments

Deduplicated Data Shipping

- Only globally unique *pages* are sent off Host
- Limited impact to network
- Reduce memory extraction time

Scan Scheduler: Building Scan Configuration

- Example:
  - X-1 bytes identical
  - Duplicate computation

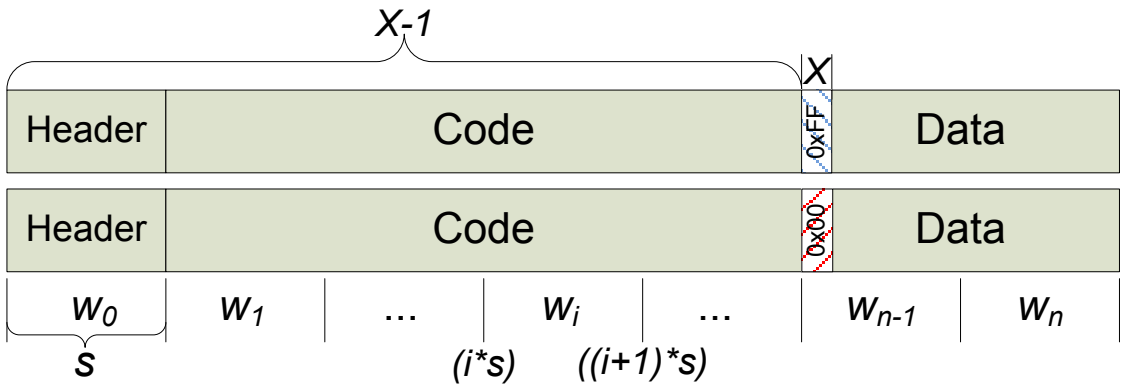


2 Similar Memory Segments: ieframe.dll

- Observation:
  1. Scanners read data linearly
  2. Scanners contain identical state for files with identical prefixes
     - Scanners duplicate computation for identical prefixes
- Goal: quickly identify prefixes across segments

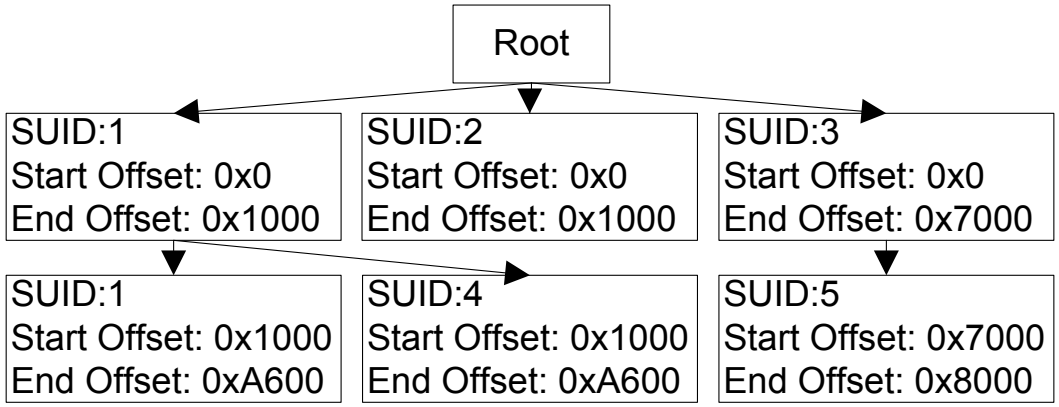
Scan Scheduler: Building Scan Configuration

- Non-overlapping window hashes
  - Data independent
  - Constrained to window sizes
  - Misses at most (window size – 1) bytes of data



Scan Scheduler: Building Scan Configuration

- Encoding Prefixes*
  - Track all unique prefixes for all *memory segments*
  - Directed Rooted Tree
    - Each node represents a range of data to scan
    - Child nodes represent different data



0x8000 bytes saved

SUID	VM	Address	Size	PID
1	1	0x112000	0xA600	872
2	1	0x60000	0x1000	872
3	1	0x1e000	0x7000	872
4	1	0x482000	0xA600	1788
5	1	0xff0000	0x8000	1788

*Algorithm and Optimizations in Paper

Prefix-Based Scanning

- Adapting a Virus Scanner
  - Modified **ClamAV**
    - Open source virus scanner
    - Only 9 lines modified
  - Shared library to hook file operations
  - **Scan tree is walked during reads**
  - **Scanner is forked at node boundaries**
  - **Ignore non-present pages**

Evaluation

- Impact of SEER on Guest VMs
- Efficiency of Prefix-Based Scanning
- Ability to find malware in memory

- Setup:
  - KVM Virtual Machines
    - Windows 7 SP1
    - 1GB Ram

Impact on Guest VMs

- Measure degradation of throughput during shipping of deduplicated data
- Specweb'2009 Banking Application
  - Simulate 200 clients

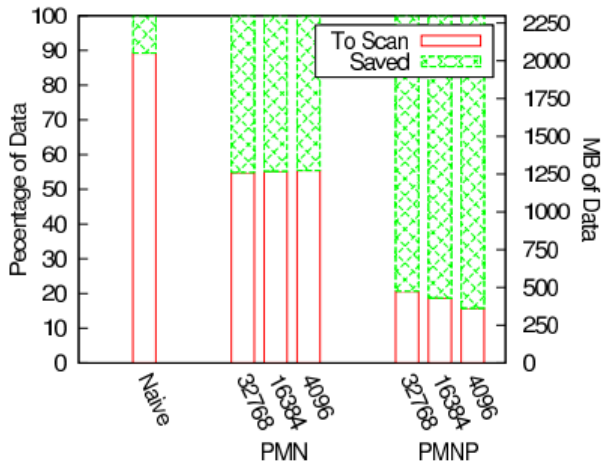
VM Impact	MB/s	Resp. Time	Snapshot Time	Data Shipped
Scanned	8.02%	9.14%	4.90 sec	153.4 MB
Non-Scanned	0.70%	0.76%	4.48 sec	140.4 MB

Efficiency of Prefix-Based Scanning

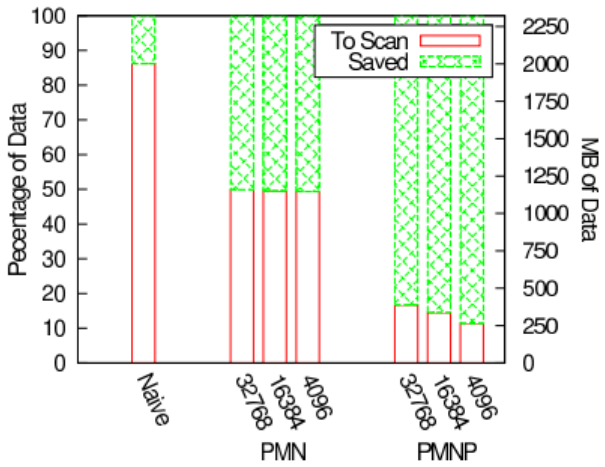
- Three Scanner configurations:
  1. Naïve – Unmodified ClamAV
  2. Prefix Matched with Normalization (**PMN**)
  3. Prefix Matched with Normalization ignoring Non-Present (**PMNP**)
- Three VM configurations:
  1. Boot to Login
  2. Webserver w/ 100 simultaneous clients
  3. Assorted GUI applications

Efficiency of Prefix-Based Scanning

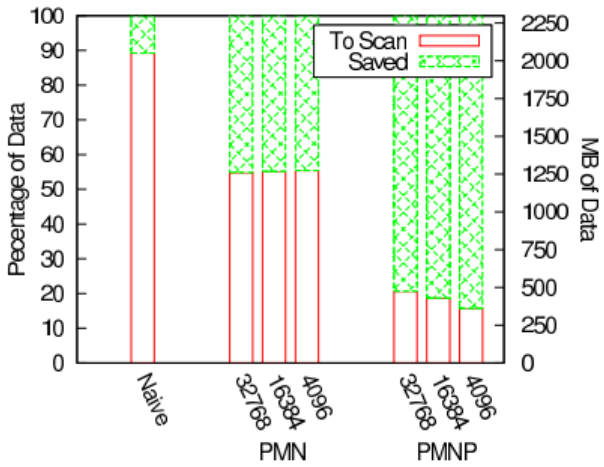
- Percent data savings and to scan
  - X-axis is window size
  - PMN (40-45%), PMNP (80-90%) reduction



(a) Boot to login



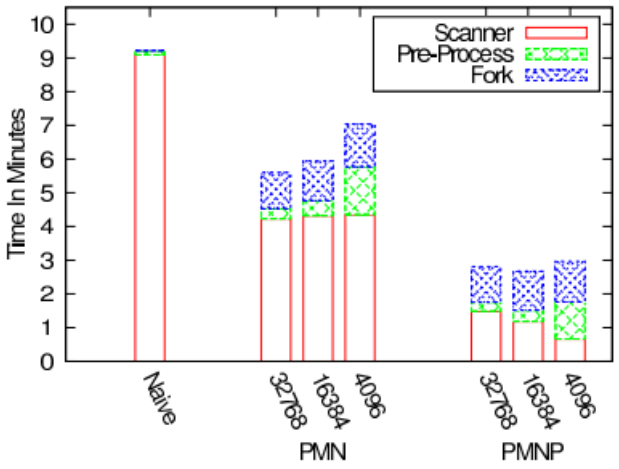
(b) Loaded IIS webserver



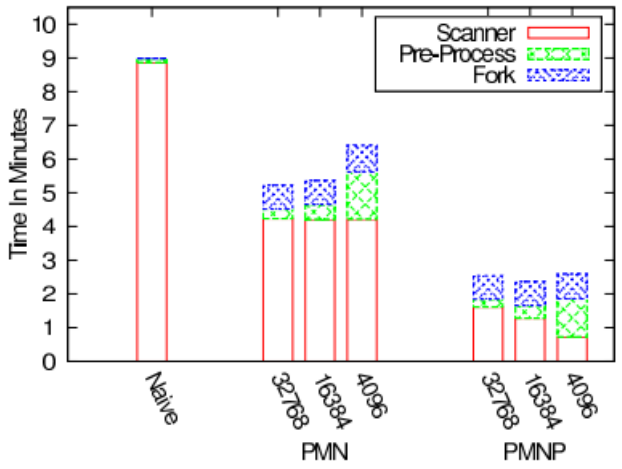
(c) Unique GUI applications

Efficiency of Prefix-Based Scanning

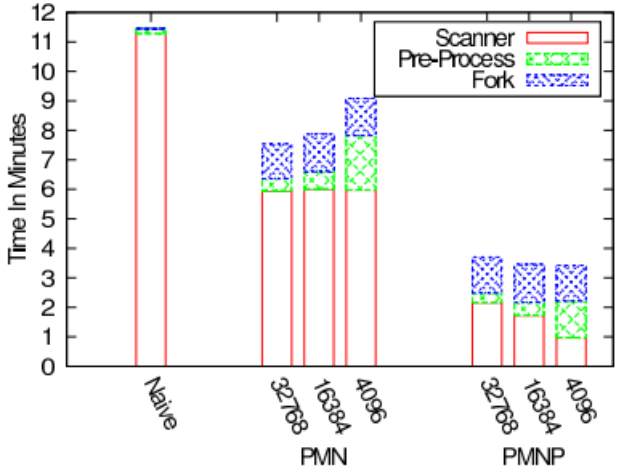
- Total scan-time for single VMs
  - X-axis is window size
  - PMN (20-42%), PMNP (62-72%) reduction



(a) Boot to login



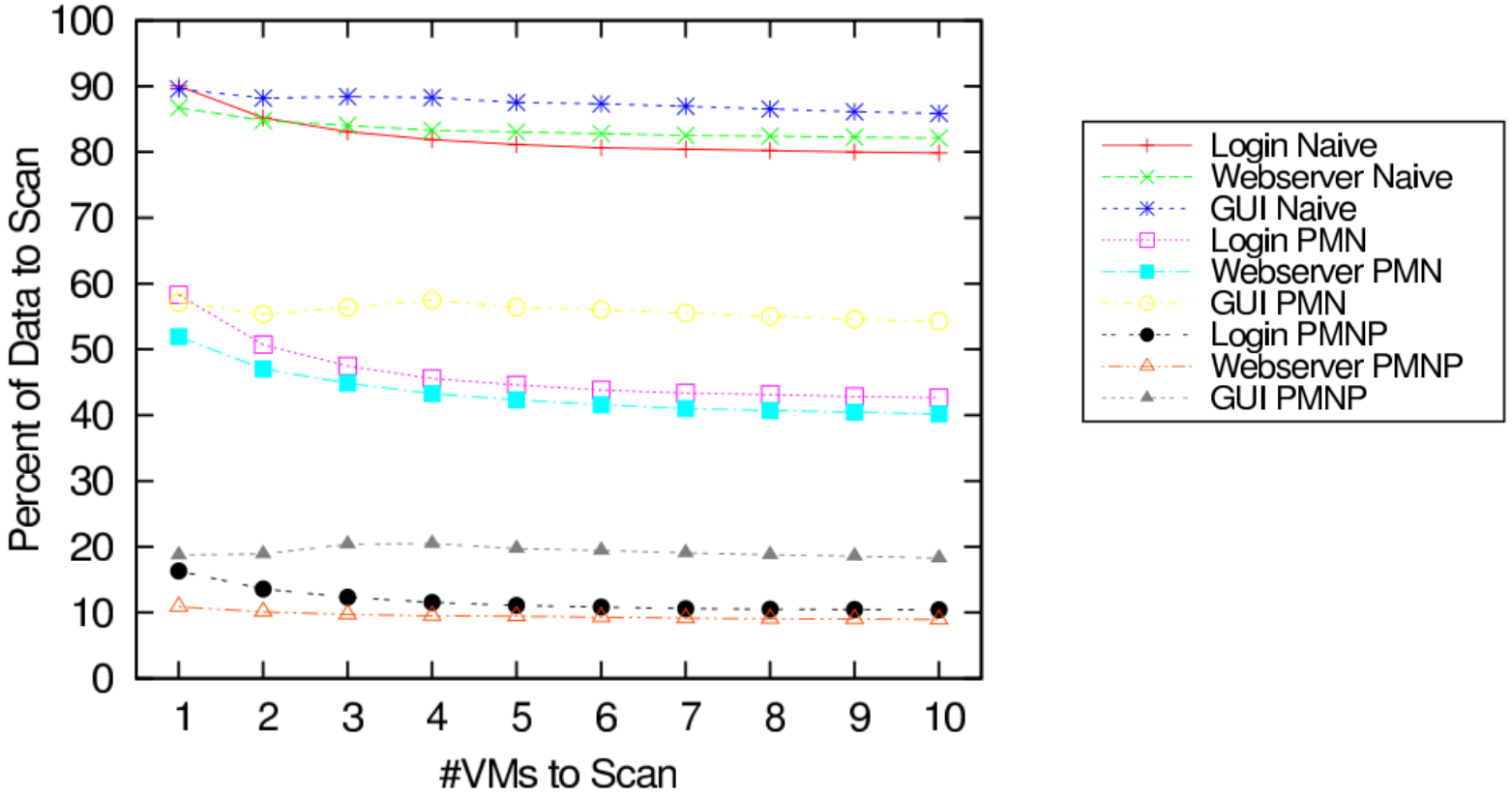
(b) Loaded IIS webserver



(c) Unique GUI applications

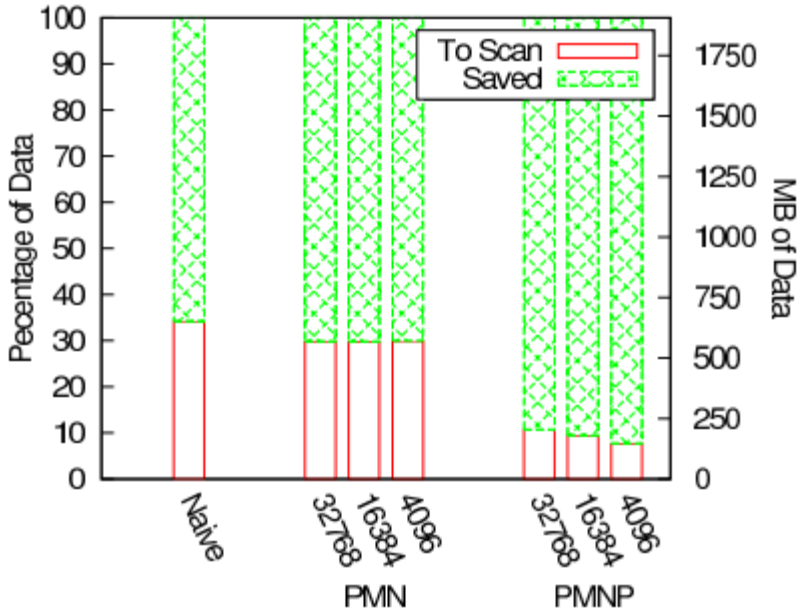
Efficiency of Prefix-Based Scanning

- Savings of scanning multiple VMs
  - Window size 32KB

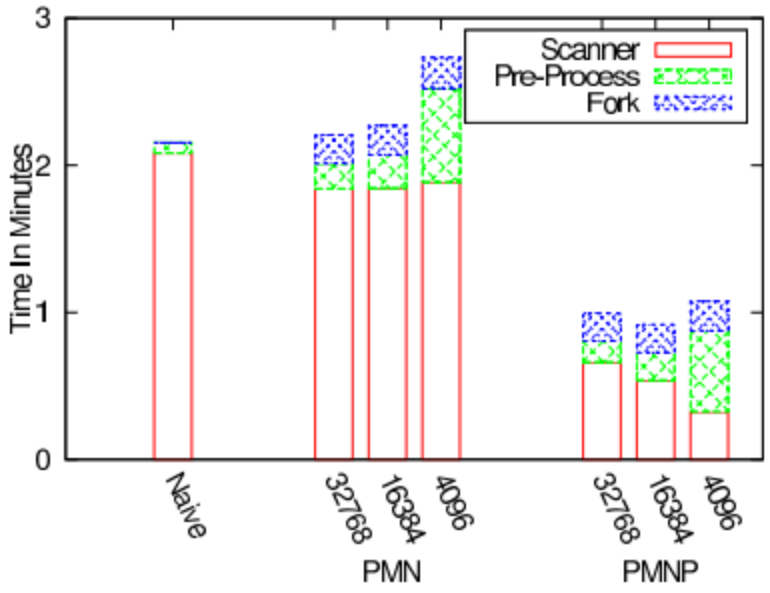


Efficiency of Prefix-Based Scanning

- Effects of caching previous scan data
  - Boot to login
  - PMNP (~50%) 32KB window



Percent Data to scan and savings with Warm Cache



Total Scan time with Warm Cache

Identifying Malware in Memory

- False Positives
  - Overly broad virus definitions
    - 154 of 2,056,340 MD5 hashes were of *zeroed* data
  - In host virus scanners
- Identified Malware
  - Memory-Only
    - Meterpreter
  - File Based
    - Cerberus RAT

Malware Types	# Samples
Trojan.Adload	2
Trojan.NSIS.Agent	7
NSIS.Clicker.Agent	1
W32.AdInstall	1
Trojan.IRC.Zapchast	1
WIN.Trojan.DarkKomet	1
Trojan.Adload	1
Adware.Cpush	1
Trojan.Spy	2
Total	17

Conclusion

- Architecture for practical Memory VSaaS
- Transparency to guest VMs
- Minimal impact to network
- Proposed new technique for scanning unique data
  - 72% reduction wall time (cold cache)
  - 50% reduction wall time (warm cache)

Thanks

- Questions?

✉ jjgionta@ncsu.edu
gionta.org