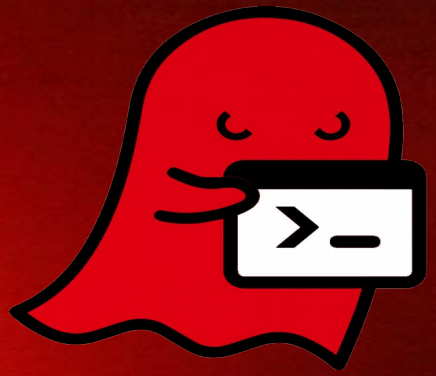
hexhive

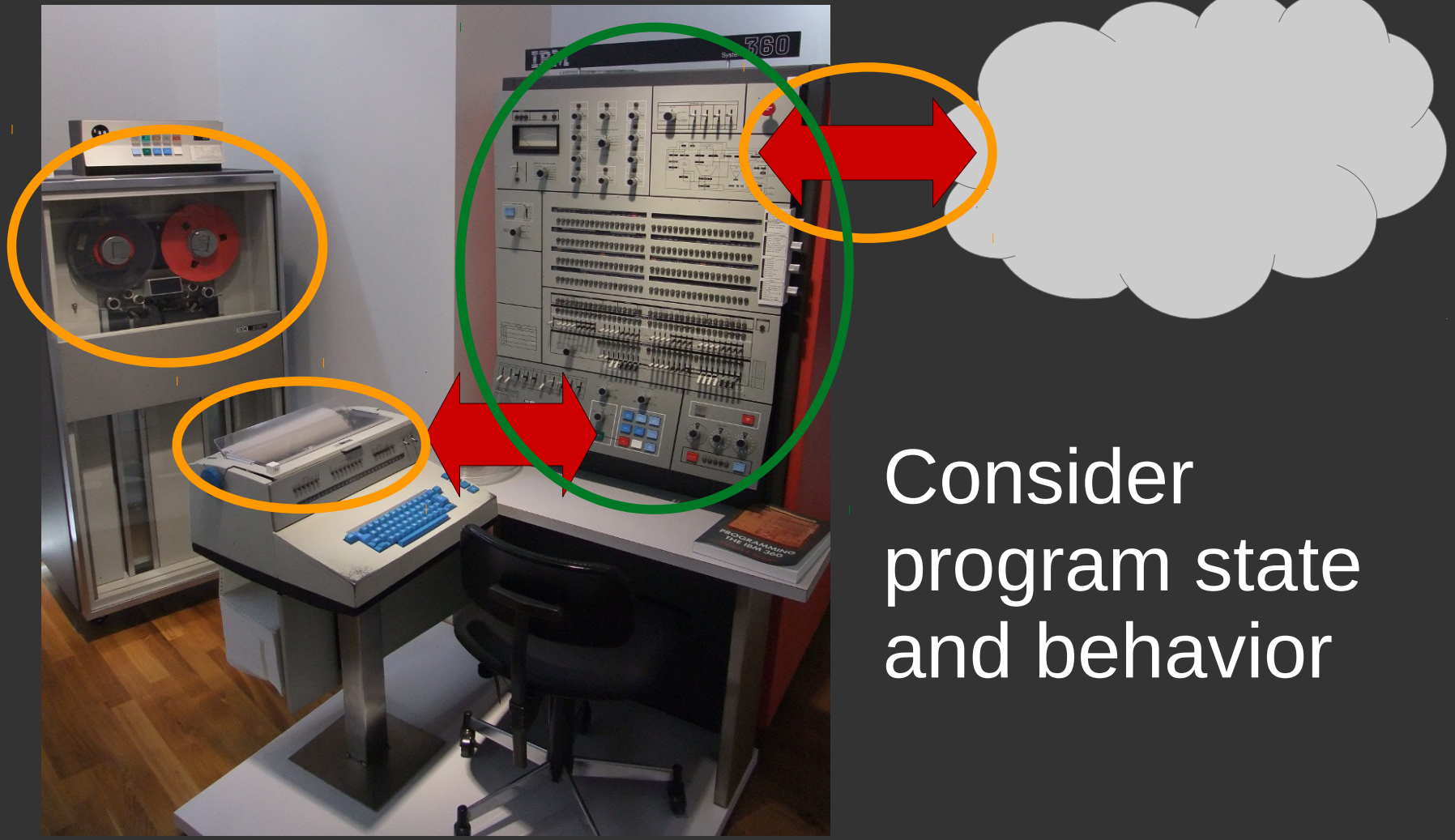
HexPADS: a platform to detect “stealth” attacks

Mathias Payer (@gannimo), Purdue University
http://hexhive.github.io

(c) AP Photo/RIA Novosti, Alexei Druzhinin, Government Press Service

Deployed defenses focus on memory corruption



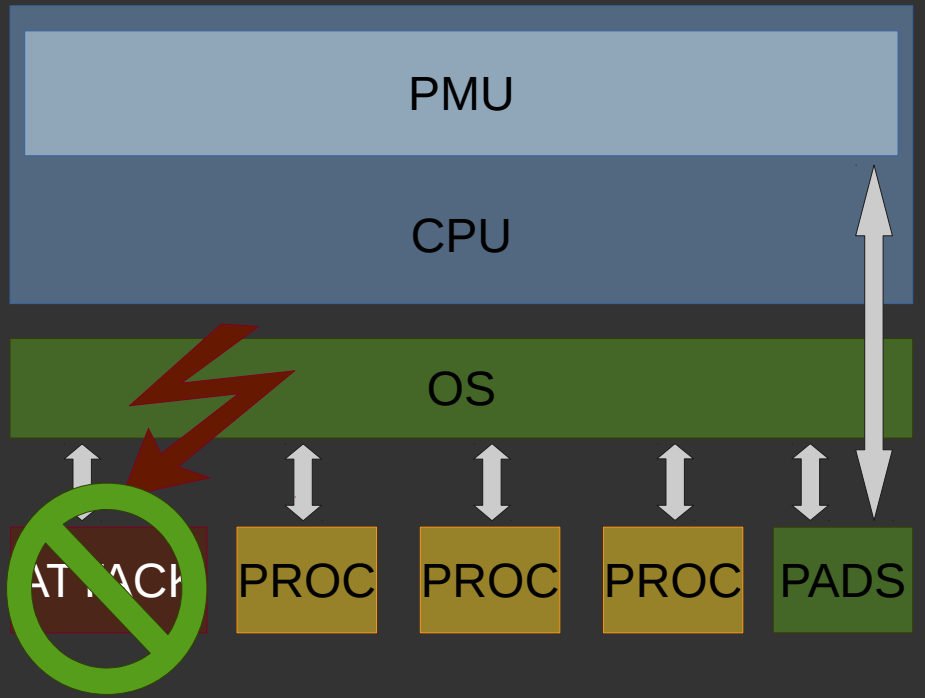


Consider program state and behavior

HexPADS Design

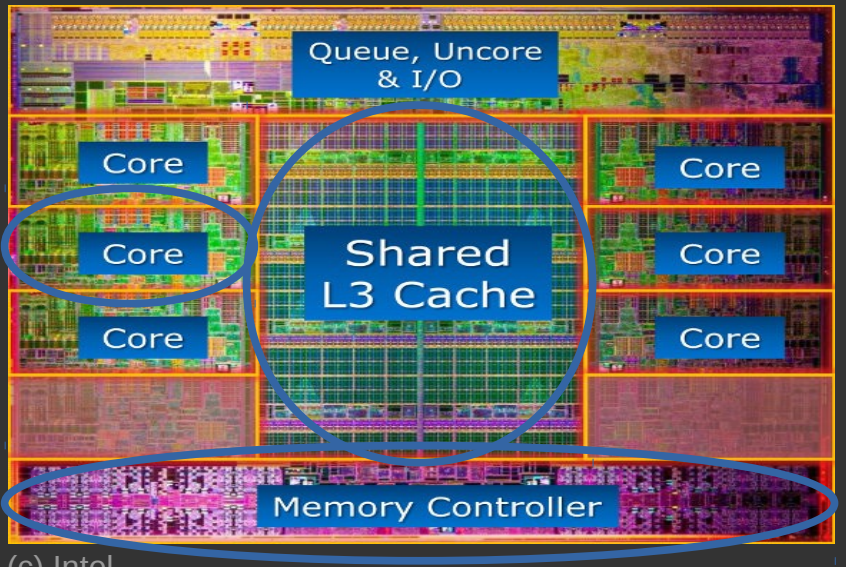
HexPADS Design

- Host-based Intrusion/Attack Detection System
- Measure fine-grained process-level ***runtime*** behavior
  - Operating system provides basic runtime characteristics
  - Performance Monitoring Unit (PMU) allows counting/sampling of detailed and fine-grained events
- Detect attacks based on signatures/anomalies
- Take evasive action/counter measure



Default Metrics (always collected)

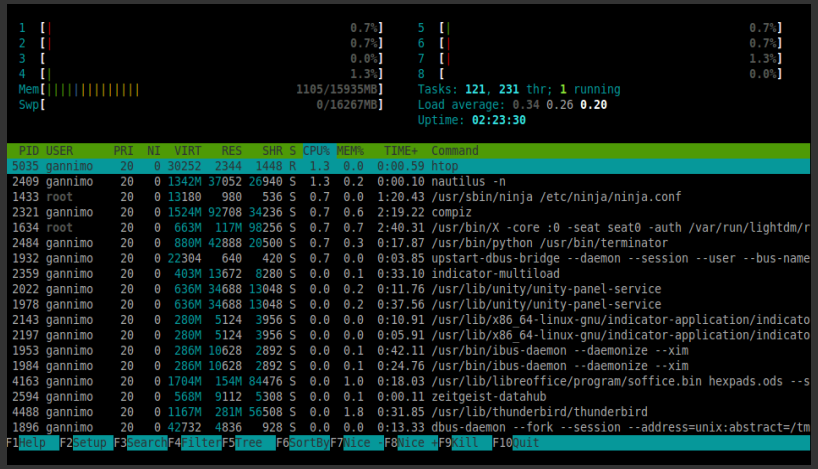
- Number of executed instructions
- Number of last level cache accesses
- Number of last level cache misses
- Minor/major page faults
- Execution time



(c) Intel

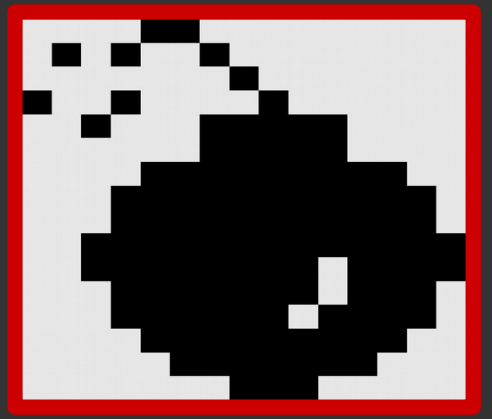
Additional Metrics

- Anything in /proc
  - Opened files, network ports, and IPC
  - Loaded libraries
  - Memory maps
- Any measurable PMU event
  - Memory/cache hierarchy events
  - Instruction mix and behavior
  - Execution profile and branch records
- System calls



Implementation

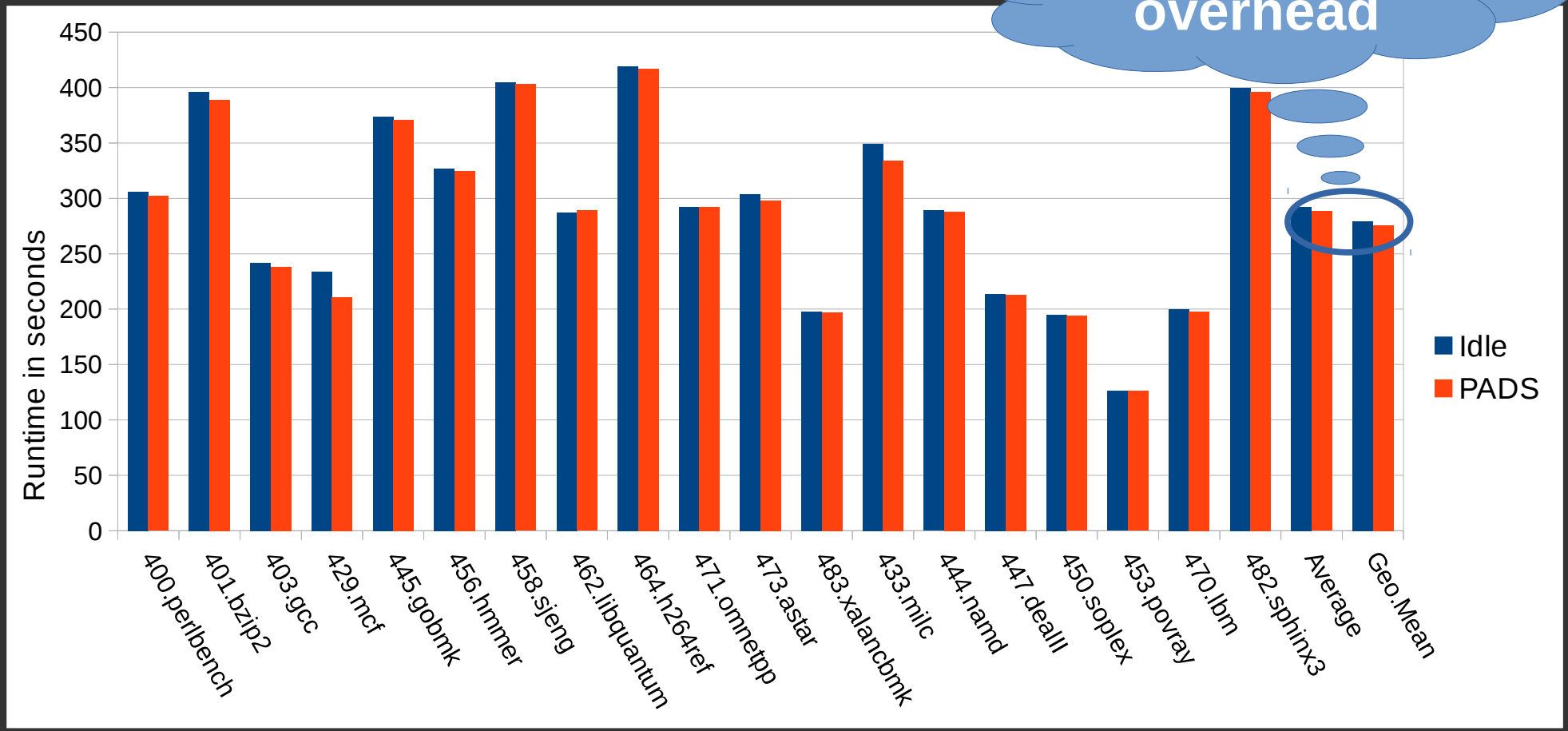
- Modular implementation
- Collect metrics for all processes
- Keep configurable history
- Run detection modules every iteration



http://github.com/HexHive/HexPads

Evaluation

SPEC CPU2006

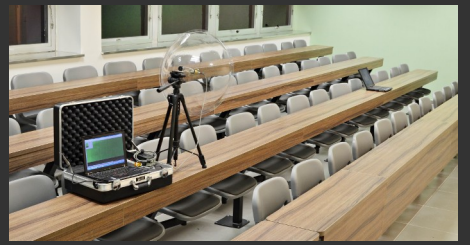


Rowhammer

- Cause DRAM bit flips by accessing adjacent cells
  - High amount of cache misses: > 500,000/s
  - High cache miss rate: > 70%
  - Low page fault rate: < 1%
- Possible extension: use sampling
  - Detect and correlate actual accesses
  - Detect “nearby” accesses

Cache-based side/covert channels

- Communicate through access timing
  - Same pattern as rowhammer
  - Additional challenge: which process is bad?
- Possible extension: longer history
  - Consider development over time



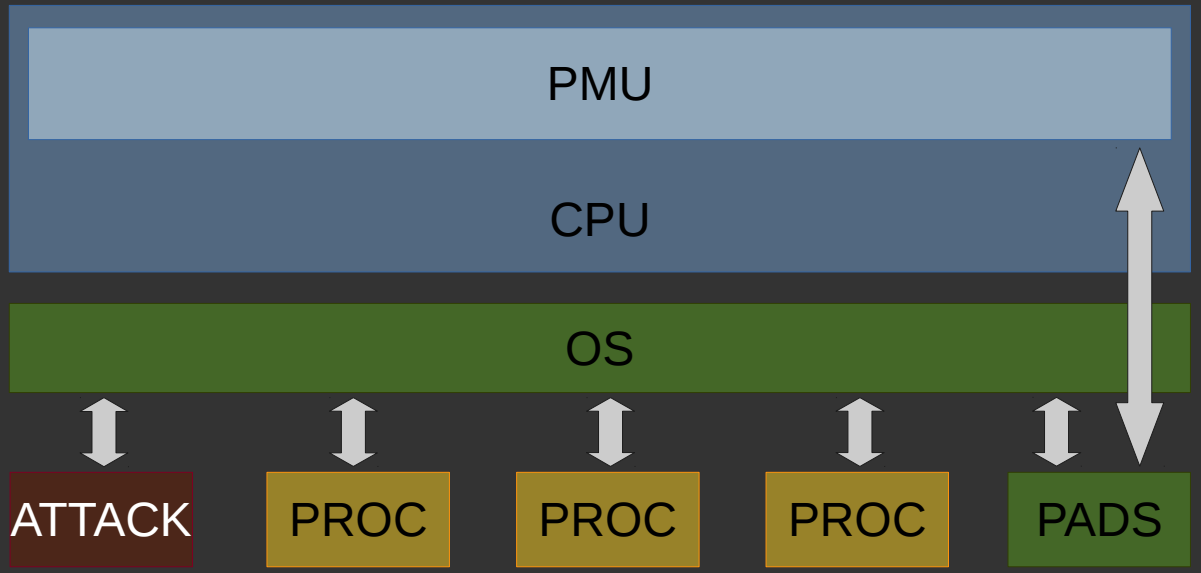
Cross-VM ASL INtrospection (CAIN)*

- CAIN attacks leak ASLR base addresses in co-located VMs
  - High amount of page faults/allocated pages/cache misses/per instr.
  - Followed by inactivity
- Possible extension: study access patterns
  - Push detection to VMM level
  - Check page similarity
  - Evaluate page access patterns

CAIN: Silently Breaking ASLR in the Cloud.
Antonio Barresi, Kaveh Razavi, Mathias Payer, and Thomas R. Gross. In WOOT '15

Upcoming Challenges

- Move collection to VMM to allow per-machine correlation
- Extend and develop new detection modules
- Synthesize detection modules by applying machine learning



Conclusion

Conclusion

- HexPADS is a modular IDS/ADS framework
- Process-based collection of runtime/performance information
- High precision and negligible overhead through PMU
- Ongoing work:
  - More detection modules
  - Machine learning
  - Push framework to VMM level
- Go clone the project at https://github.com/HexHive/HexPADS

hexhive

Thank you!
Questions?

Mathias Payer (@gannimo), Purdue University
http://hexhive.github.io