Scalable Security Vulnerability Analysis via Sampling

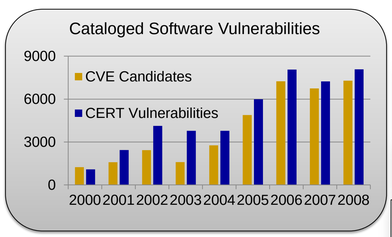
Joseph Greathouse, Ilya Wagner, Valeria Bertacco, and Todd Austin, University of Michigan

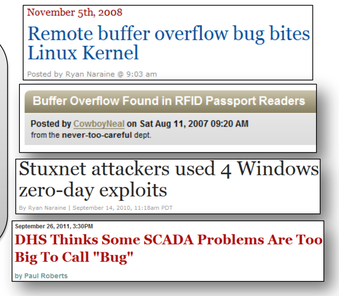
Resilient Systems, 5.5.7

GSRC Annual Symposium
Nov 16, 2011
Pub ID 2654

Motivation: Software Errors Abound

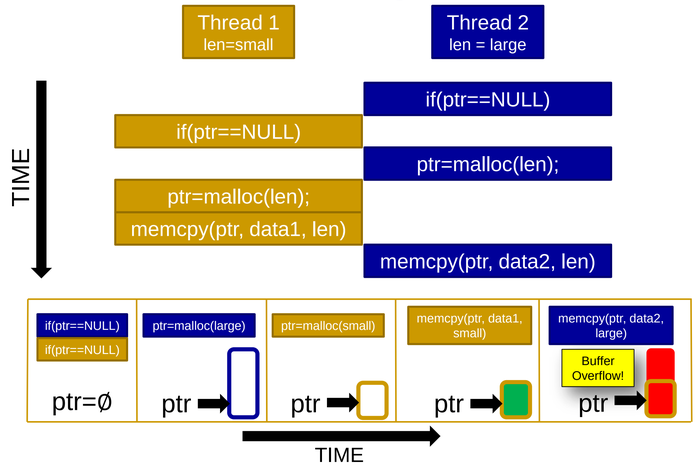
- NIST: SW errors cost US **~$60 billion/year** as of 2002
- FBI CCS: Security Issues **$67 billion/year** as of 2005
  - >⅓ from viruses, network intrusion, etc.





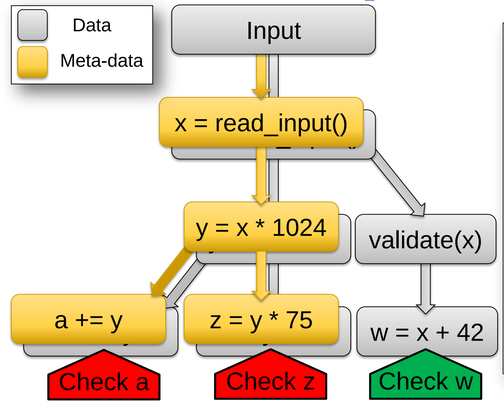
1

A Modern Security Bug



2

Dynamic Dataflow Analysis Techniques are Powerful but Expensive



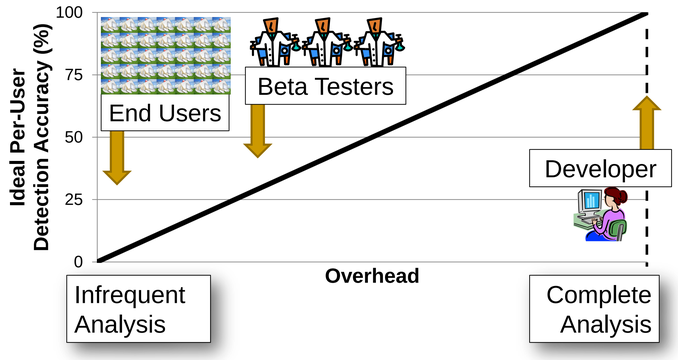
Overheads

Memory Checking	5-50x
Taint Analysis	2-200x
Data Race Detection	2-300x
Symbolic Execution	10-200x

3

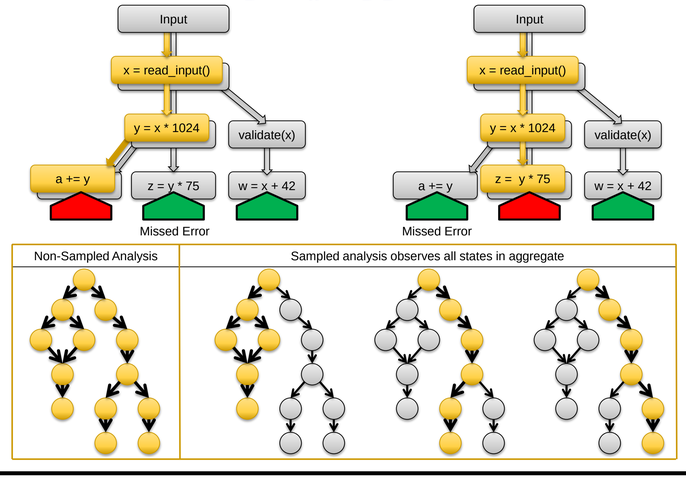
Our Solution: Dataflow Sampling

Lower overheads by skipping some analyses



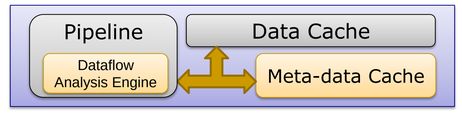
4

Dataflow Sampling Approach

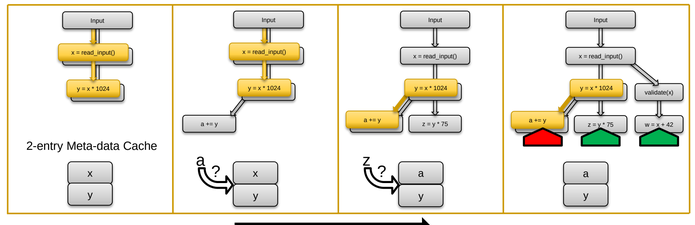


5

Testudo: Hardware Dataflow Sampling



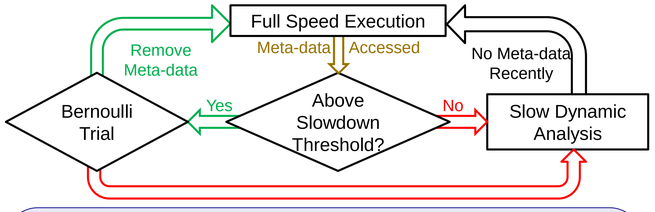
- What happens when the cache overflows?
  - **Use sampling to randomly remove meta-data**
  - Reduces off-chip changes while facilitating meta-data sampling

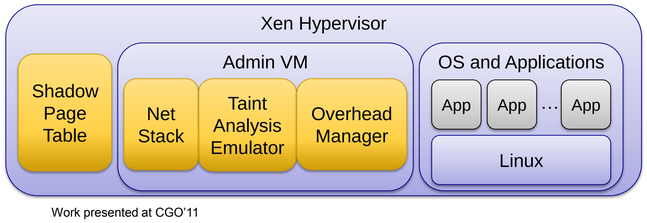


Work presented at MICRO'08

6

Software-Based Dataflow Sampling

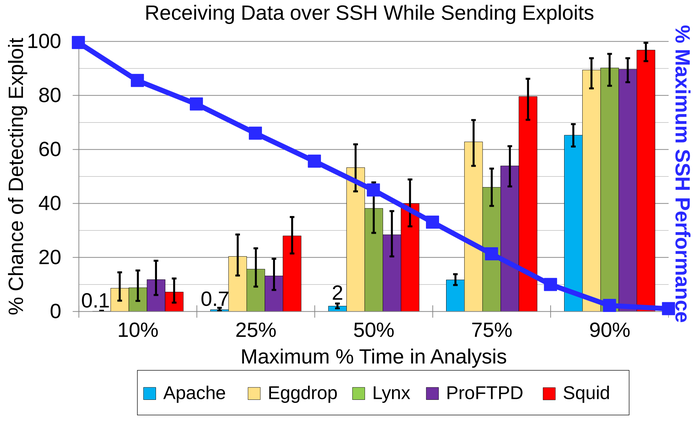




Work presented at CGO'11

7

Accuracy and Performance



8

Other Applications and Future Work

- Generalized hardware mechanism to find meta-data
  - Use hardware-supported watchpoints to speed up many analyses
  - To be presented at ASPLOS'12
- Simple hardware to accelerate data race detection
  - Performance counters enable SW race detector on-demand
  - Presented at ISCA'11
- Use dynamic sampling to improve static software analysis
  - Control paths near executed code are often exploited by hackers
  - Keep track of paths users don't execute across entire population
  - Focus on these paths in static analysis tool

9