Avfs

An On-Access
Anti-Virus File System

Yevgeniy Miretskiy, Abhijith Das,
Charles P. Wright, and Erez Zadok
Stony Brook University

http://www.fsl.cs.sunysb.edu/

Design Goals

- Accuracy
  - True on-access scanning
  - File-system modes
- Performance
  - State-oriented design
  - Scalable in-kernel virus scanner
- Transparency
  - Stackable file system
  - Layers on top of Ext2, NFS, etc.

Accuracy

- Most virus scanners scan: on-**`open`**, on-**`close`**, or on-**`exec`**
- We scan on-access: read and write
  - Read: before data is returned to the user
  - Write: before data is committed to stable storage

File System Modes

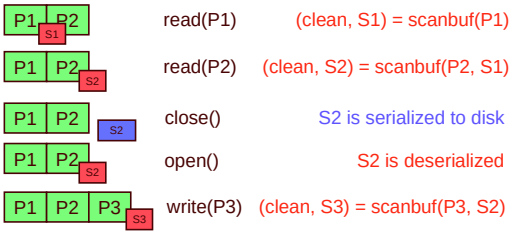
- Immediate
  - When a virus is written, the write fails before the data is committed to stable storage
  - When a virus is read (e.g., existing file system) it is quarantined
- Forensic
  - A version is created on the first write
  - No error is returned until entire virus is written
  - The file is reverted to the good copy on close
  - Quarantine and save infected file with extra info

Performance

- We built on ClamAV OSS scanner
  - Aho-Corasick builds automaton to quickly find many patterns in an input
  - ClamAV scales poorly with no. of signatures: O(*n*)
- Our improved scanner is called **Oyster**
  - Variable height tries: scales as O(*lg n*)
  - Kernel-based scanner: reduce copies
  - Allow administrators to trade memory for speed
  - 4.5x faster than ClamAV: 1GB file, 20K sig's.
- Avfs overhead over Ext2 (using Oyster)
  - 14.7% for ~20K signatures
  - 52.3% for 128K signatures

Stateful Scanning

- Scanner is separate from the file system
- We scan one page at a time, but viruses can span multiple pages



Status

- Oyster Scanner
  - Runs in-kernel, exports generic API
  - Scales logarithmically
- Avfs
  - On-access scanning
  - State-oriented
- Future Work
  - Per-page state for multi-part viruses
  - Improved string matching for leaf nodes

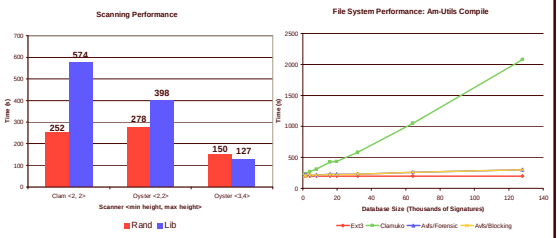
Questions?

Avfs:An On-Access Anti-Virus File System

Yevgeniy Miretskiy, Abhijith Das, Charles P. Wright, and Erez Zadok
Stony Brook University

To Appear in:
USENIX Security Symposium '04

Performance



- Oyster <3,4>: 4.5 times faster than ClamAV
- Avfs
  - 14.7% overhead for 19.8K (actual database size)
  - 52.3% for 128K signatures

Scanning Trie

- Aho-Corasick Algorithm
- ClamAV: Fixed trie height of two
  - 10% of 2-character prefixes make up 57% of patterns
  - Up to 700 patterns per 2 character prefix
- Oyster:Variable trie height