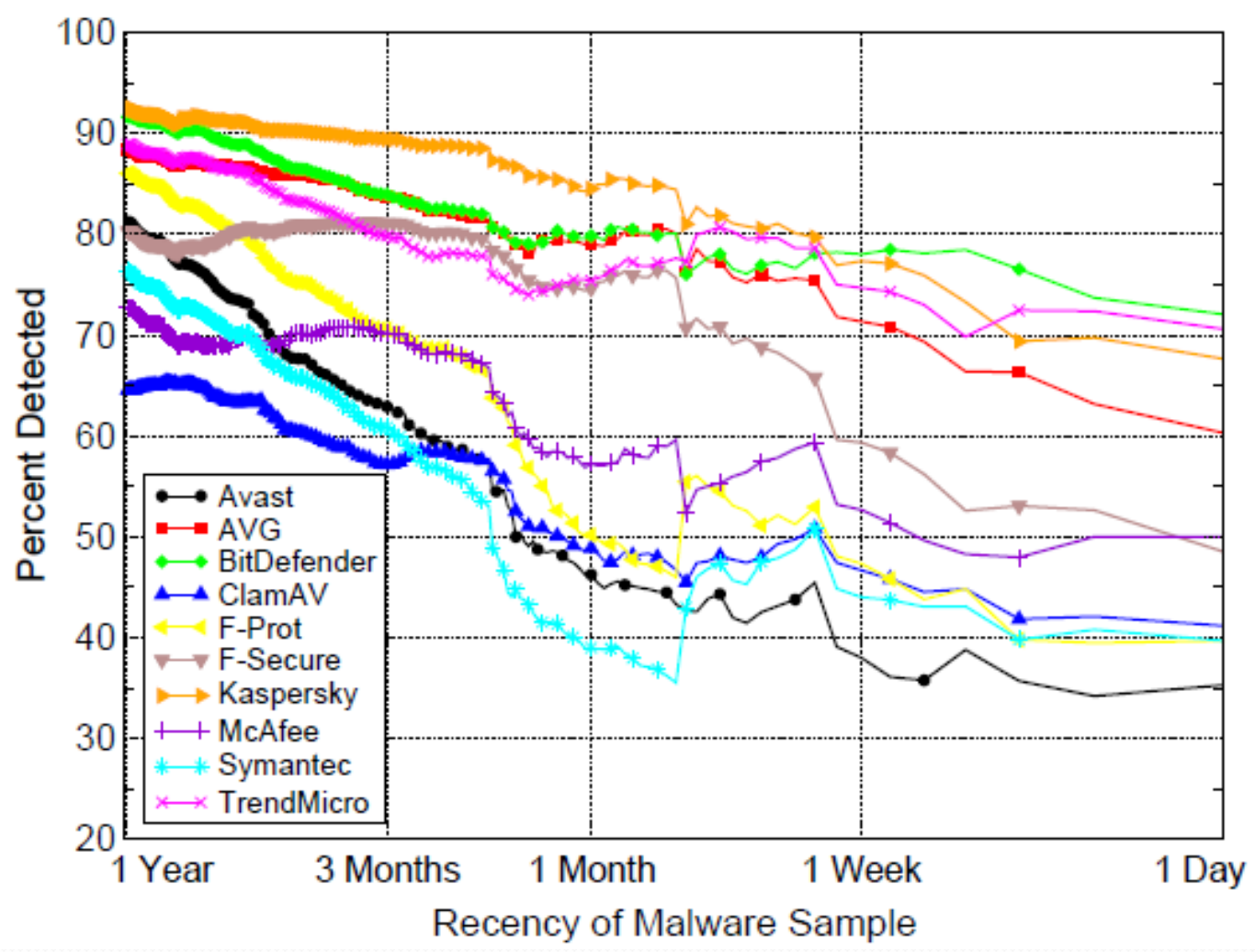
CloudAV: N-Version Antivirus in the Network Cloud

Presentation by Brett Meyer

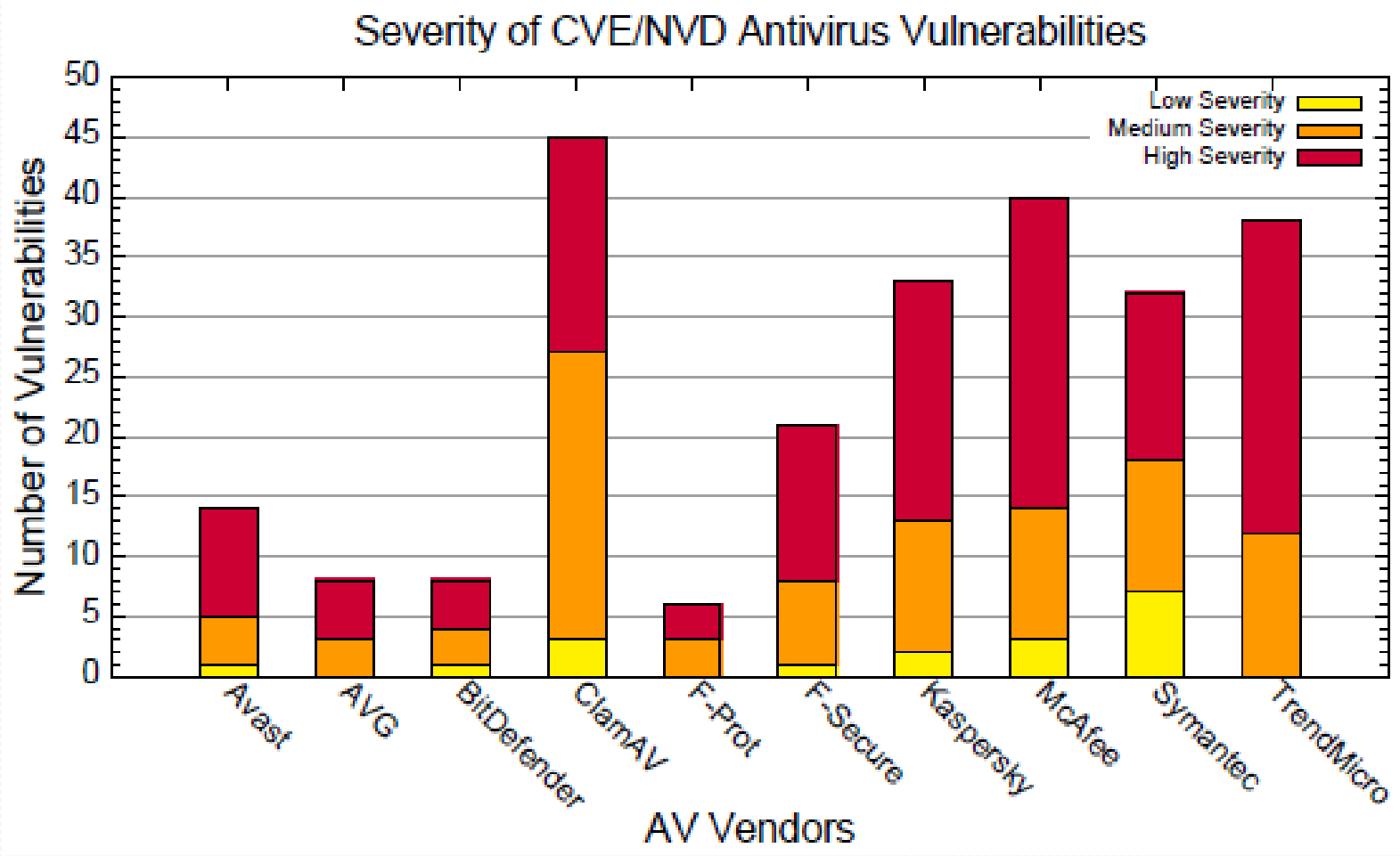
Traditional AV Software Problem 1: Signature generation

- Signature based detection model
- Sheer volume of new threats limits number of signatures created by one vendor
- Not good for zero-day malware, vulnerability window too great
- Detection rates can drop over 45% when comparing malware that is a year old versus malware that is a day old


100
90
80
70
60
50
40
30
20
Percent Detected
Avast
AVG
BitDefender
ClamAV
F-Prot
F-Secure
Kaspersky
McAfee
Symantec
TrendMicro
1 Year
3 Months
1 Month
1 Week
1 Day
Recency of Malware Sample

Traditional AV Software Problem 2: Complexity

- As the complexity of AV software increases, so do its vulnerabilities
- Local and remote exploits of AV software have been observed in the wild
- Since AV software needs elevated user privileges to operate, vulnerabilities lead to a complete compromise of end host machines


Severity of CVE/NVD Antivirus Vulnerabilities
Number of Vulnerabilities
0
5
10
15
20
25
30
35
40
45
50
Low Severity
Medium Severity
High Severity
Avast
AVG
BitDefender
ClamAV
F-Prot
F-Secure
Kaspersky
McAfee
Symantec
TrendMicro
AV Vendors

The answer, CloudAV!

- Two major principles:
  - Antivirus as a network service
    - Analysis of malware done as an in-cloud network service
  - N-version protection
    - Uses multiple, heterogeneous detection engines in parallel

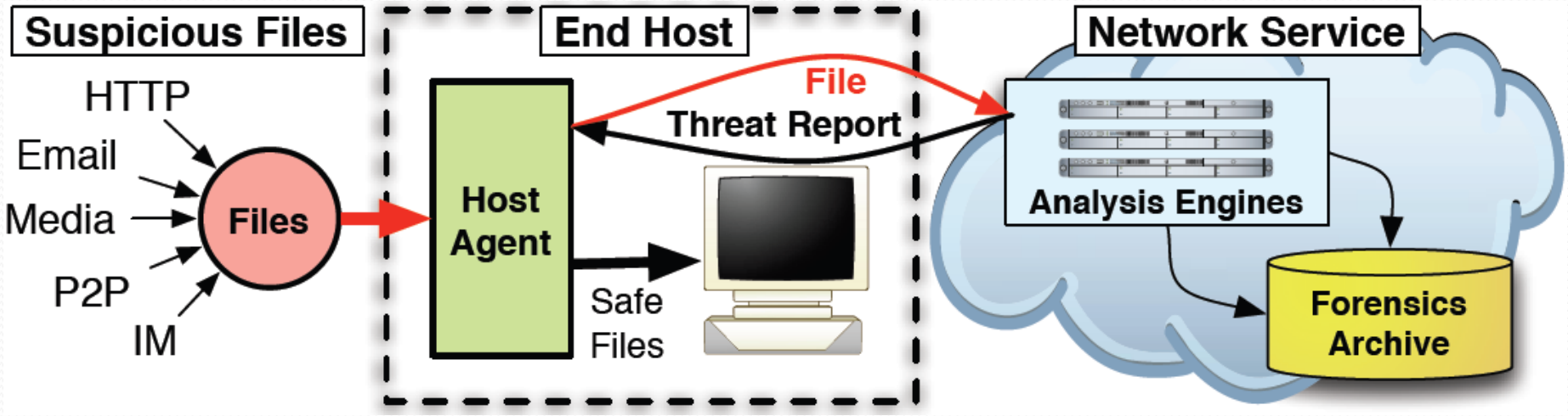
CloudAV to the rescue!

- Major benefits of this model:
  - Better detection of malicious software
  - Enhanced forensics capabilities
  - Retrospective detection
  - Improved deployability and management

The makeup of CloudAV

- Three major components:
  - A lightweight *host agent* run on end hosts
    - Designed for multiple platforms, including Windows, Linux, and FreeBSD
  - A *network service* that receives files from hosts and identifies unwanted or malicious content
    - Consists of ten antivirus engines and two behavioral detection engines
  - An *archival and forensics service* that stores information about analyzed files and provides a management interface for operators

The CloudAV model



More benefits

- Offloading the analysis tasks to the network service reduces the complexity of the host end software

- Devices like mobile phones that have limited computing power can more effectively identify malware

A quick disclaimer

- CloudAV will not replace existing antivirus or intrusion detection solutions
- Simply an extra layer of protection for environments such as enterprise networks, government networks, and mobile networks
- User files must be shipped to another computer for analysis, so privacy must be controlled and maintained in the deployment environment

Architecture: Client Software

- Incoming files are trapped and diverted to a handling routine which creates a unique identifier (UID) and compares it to previously analyzed files
- If no UIDs match, the file is shipped to the network service for analysis
- UIDs are created by cryptographic hashing since this method is fast and effective
- By reducing the complexity of the host agent, fewer attacks are possible

Architecture: Client Software

- User interface has three modes
  - Transparent mode
    - Files sent to the cloud for analysis, but execution of a file is never blocked
    - Users may become infected, but admins can use detection alerts
  - Warning mode
    - Access to a file is blocked until an *access directive* is returned to the host agent
    - Users then make a decision whether to proceed in accessing the file based on a prompt if the file is suspicious
  - Blocking mode
    - Access to a file is blocked until an *access directive* is returned to the host agent, and then access to suspicious files is denied

Architecture: Network Service

- Each file is analyzed by multiple detection engines in parallel and then a final determination is made about whether the file is malicious
- These results are aggregated into a threat report
- Additional detection engines can be added easily
- Files are analyzed quickly on a cluster of servers
- Antivirus engines and behavioral analyzers like sandboxes or VMs can be employed to make determinations about files
- Host agent files are the primary means of file acquisition, but other methods like network sensors or stream taps using DPI may also be implemented

Architecture: Network Service

- During result aggregation, a subset of results may be used due to timing constraints
- Data may also be wrapped in a container object that describes how the data should be interpreted
- The threshold at which a candidate file is deemed unsafe is set by the network administrators
- The aggregation process results in a threat report sent to the host agent, the contents of the report vary based on the deployment environment
- Threat reports are cached on the host agent and the network server for future detection

Architecture: Archival and Forensics Service

- Provides information on file usage across participating hosts
- Consists of file access information as well as behavioral information
- Amount of information is tunable by network administrators
- Allows for retrospective detection, which makes identifying zero-day software easier

Implementation

- Host agent implemented for Windows 2000/XP/Vista, Linux 2.4/2.6, and FreeBSD 6.0+
- Also implemented as a mail filter for mail transfer agents
- Communication between the host agent and the network service uses a HTTP wire protocol protected by mutually authenticated SSL/TLS
- Network service allows for prioritized analysis

Implementation

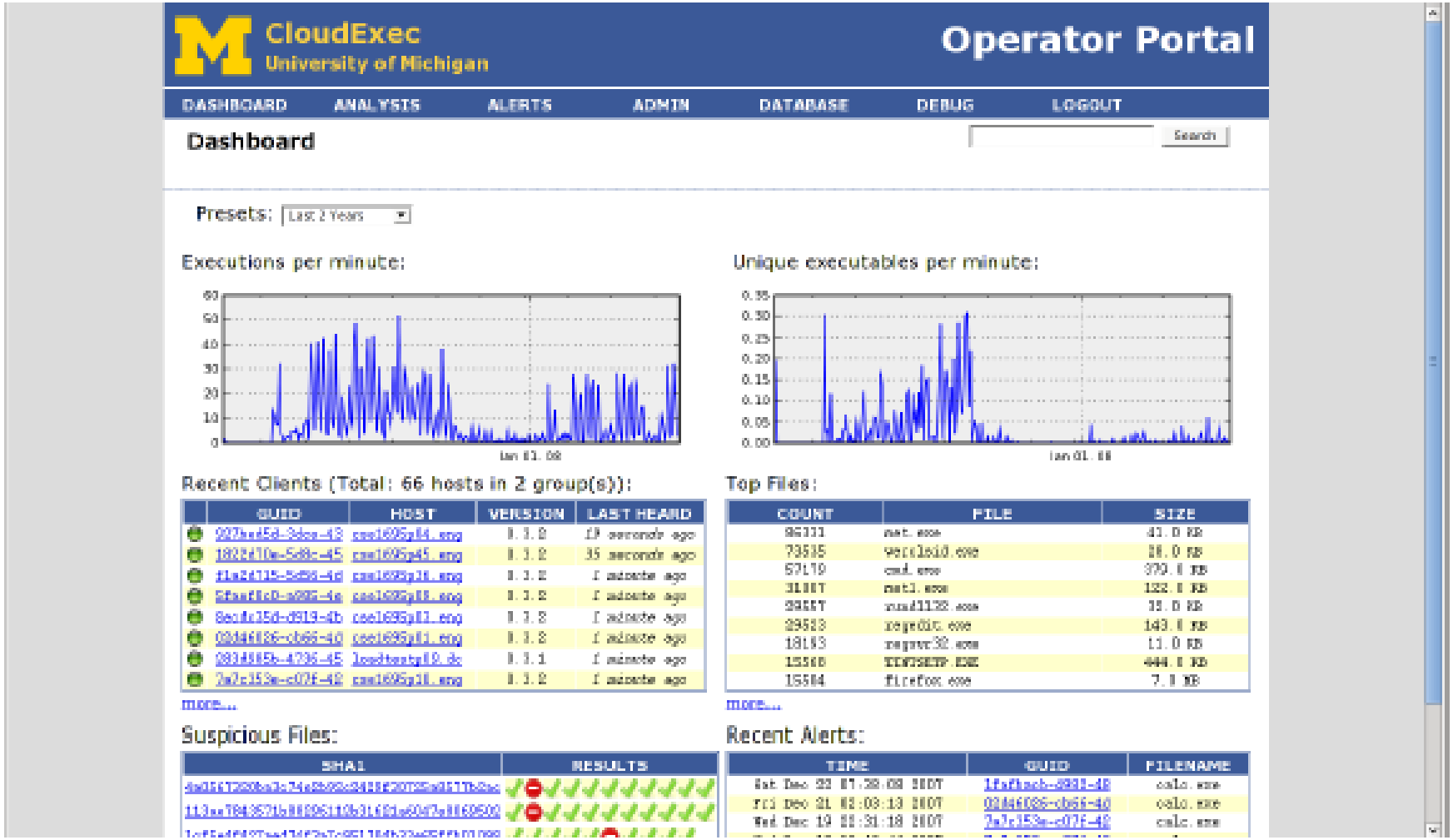
- Each backend engine runs in a Xen virtualized container for scalability, and to prevent attacks/failures of individual AV engines
- 12 engines used
  - 10 AV engines
    - Avast, AVG, BitDefender, ClamAV, F-Prot, F-Secure, Kaspersky, McAfee, Symantec, and Trend Micro
  - 2 behavioral engines
    - Norman Sandbox and CWSandbox

Implementation

- A management interface provides access to the forensics archive, policy enforcement, alerting, and report generation
- Allows for network administrators to enforce network-wide policies and define alerts when those policies are violated
- Alerts are defined through a specification language similar to an SQL WHERE clause

Detection engine VM monitoring interface

Web management portal

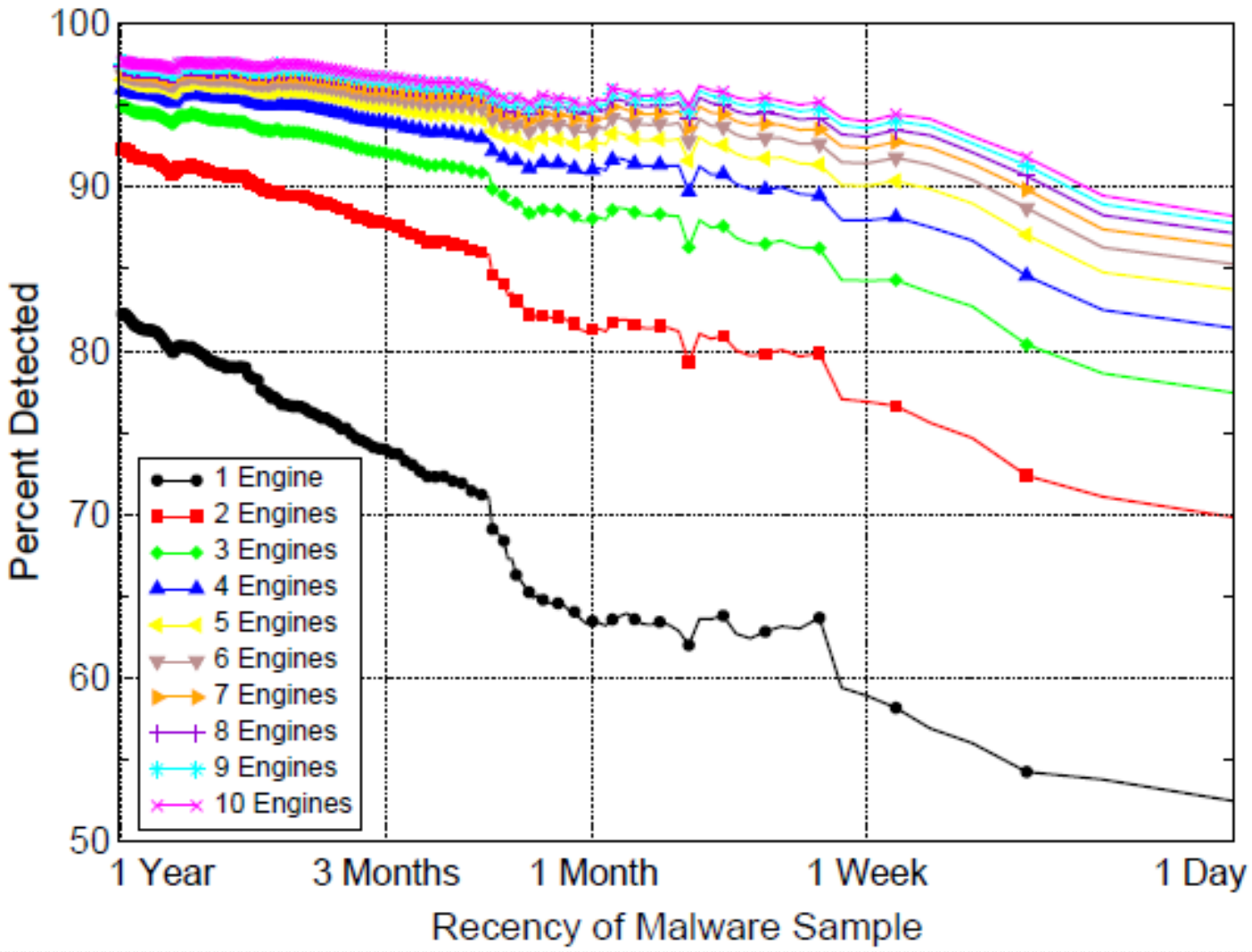


Evaluation

- Datasets
  - Evaluation of N-version protection and retrospective detection
    - 7220 malware samples collected from November 2006 to November 2007, taken from Arbor Network's Arbor Malware Library
  - Evaluation of peformance
    - Results from deploying the CloudAV system on a campus network for over 6 months

Results

- Detection rates determined by the average performance across all combinations of N engines
- Using 10 engines increases the detection rate for the year-long dataset as high as 98%
- With a single antivirus engine, detection degrades from 82% against a year old dataset to 52% against a day old dataset
- Using ten antivirus engines, performance degrades from 98% for the year-old dataset to 88% for the day old dataset


100
90
80
70
60
50
Percent Detected
1 Engine
2 Engines
3 Engines
4 Engines
5 Engines
6 Engines
7 Engines
8 Engines
9 Engines
10 Engines
1 Year
3 Months
1 Month
1 Week
1 Day
Recency of Malware Sample

Results

- Also used the AML dataset to discover the importance of retrospective detection
- Used a year's worth of McAfee DAT signature files for comparison
- Found that about 100 new malware samples were detected each week
- The average time from when a piece of malware was observed until it was detected was 48 days using McAfee

Deployment Results

- Total number of executables was about 20,500 per day
- Number of unique executables was about 217 per day
- Cache hit rate for the host agents was about 99.8%
- 2 case studies from real-world deployment
  - Malware case study
    - CloudAV correctly identified a malicious binary hidden in a keygen executable
  - Legitimate case study
    - CloudAV flagged an executable as suspicious which the network administrators were able to dismiss as a legitimate program

Limitations

- An in-cloud system can provide additional context to their detection engines through simulating the end host environment for more accurate detection
- However, the end host state may be quite large and some manner of detection engine may be needed at the host agent
- Any network disconnectivity results in the host agent being unable to access the network cache of signatures
- The deployed system focused on executables, but the system would need to be extended to include other file types, e.g. DLLs

Limitations

- Licensing for AV software can be expensive for many systems
- Using only four free AV engines (AVG, Avast, BitDefender, and ClamAV) detection rates of 94.3%, 92%, and 88% were possible for periods of 3 months, 1 month, and 1 week, respectively
- The number of false positives increases with the number of engines used
- Aggregating results from multiple engines and using thresholds or centralizing the network administration mitigates this side effect

One last benefit!

- CloudAV is innately vendor-neutral, and it offers organizations an opportunity to break free of vendor lock-in

Questions?