CloudAV

N-Version Antivirus in the Network Cloud

Jon Oberheide, Evan Cooke, Farnam Jahanian

University of Michigan

July 30, 2008

USENIX Security '08

Roadmap

- **Motivation and Limitations of Antivirus**
- AV as an In-Cloud Network Service
- Implementation and Evaluation
- Discussion and Wrap-up

Antivirus

Antivirus is the predominant method of detecting and stopping malicious software

- Widely deployed
- Last line of defense
- Over $10 billion market in 2008
- Over 50% of security software revenue

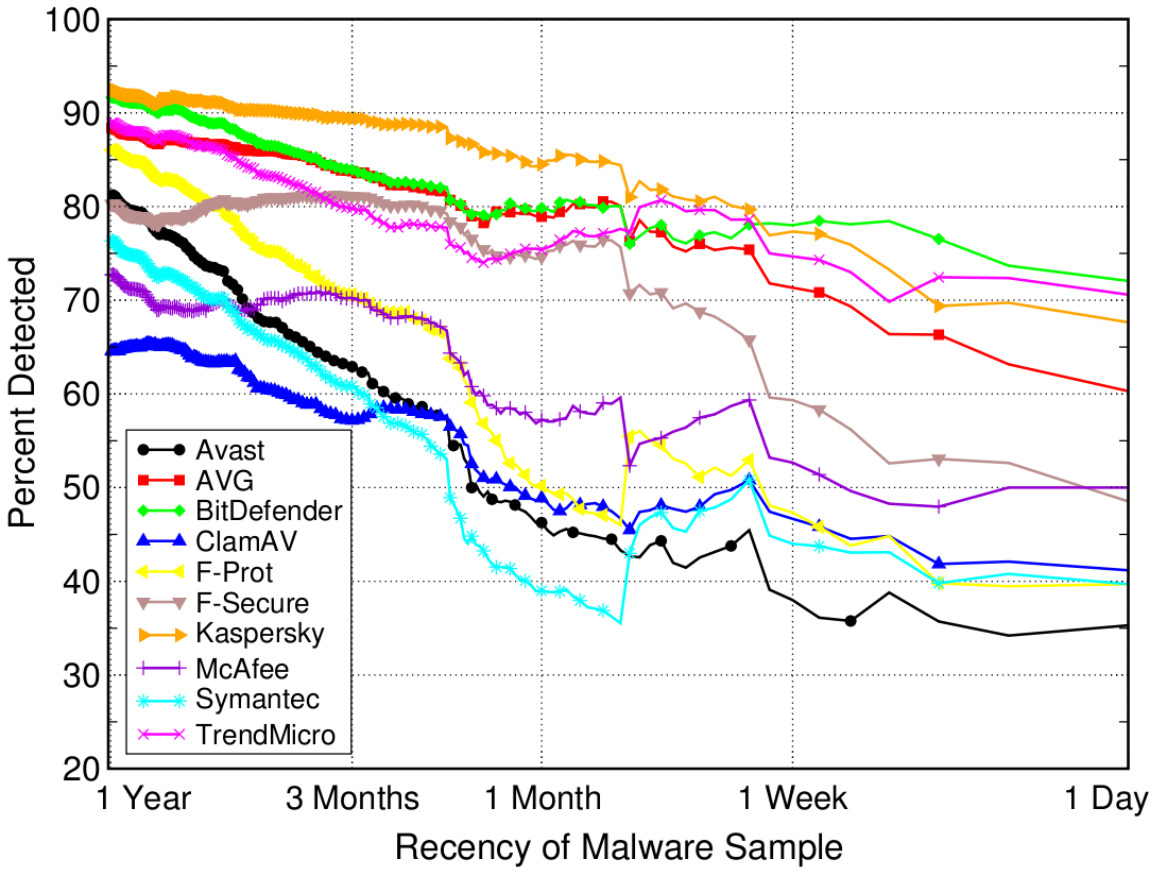
Antivirus Limitations

- Detection Coverage
  - Dismal detection rates
  - Slow response to emerging threats
  - Disjoint detection/collection methods
- AV Software Vulnerabilities
  - Complexity → security risk
  - Local and remote exploits
  - Inherently high privileges

Antivirus	Detected
Avast	**45.8%**
ClamAV	**48.8%**
F-Prot	**49.6%**
F-Secure	**74.4%**
Kaspersky	**84.0%**
McAfee	**56.7%**
Symantec	**38.8%**
Trend Micro	**74.6%**

Arbor Malware Library (AML)
Oct '07- Nov '07

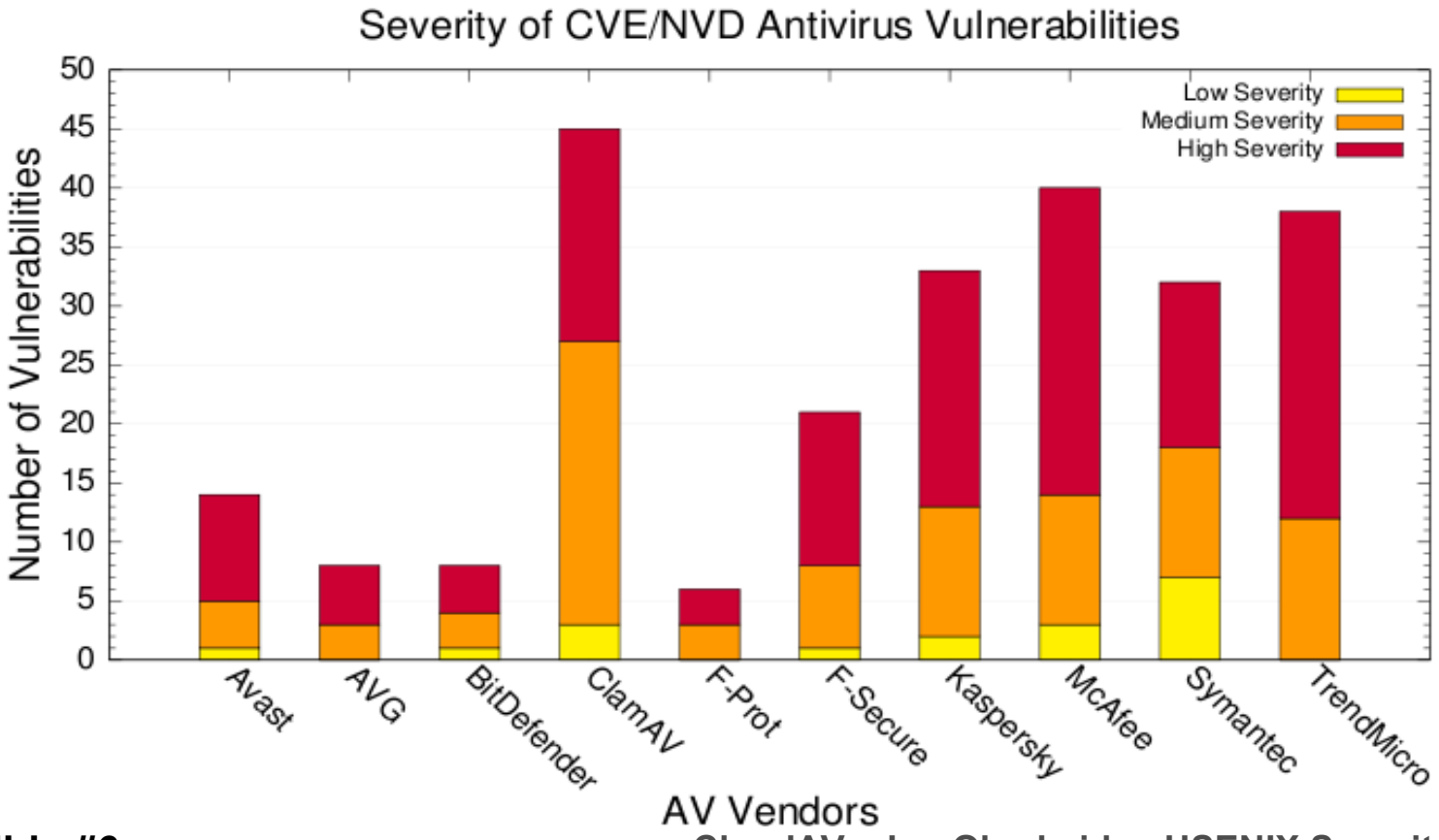
Detection Degradation



Antivirus detection coverage degrades significantly as threats approach 0-day

AV Software Vulnerabilities

Antivirus software is listed as one of the top 20 threats of 2007 according to SANS



Addressing the Limitations

- Detection Coverage
  - Dismal detection rates
  - Disjoint detection/collection methods

Leverage detection capabilities from multiple vendors

- AV Software Vulnerabilities
  - Complexity leads to security risk
  - Inherently high privileges

Need isolation between end host and analysis engines

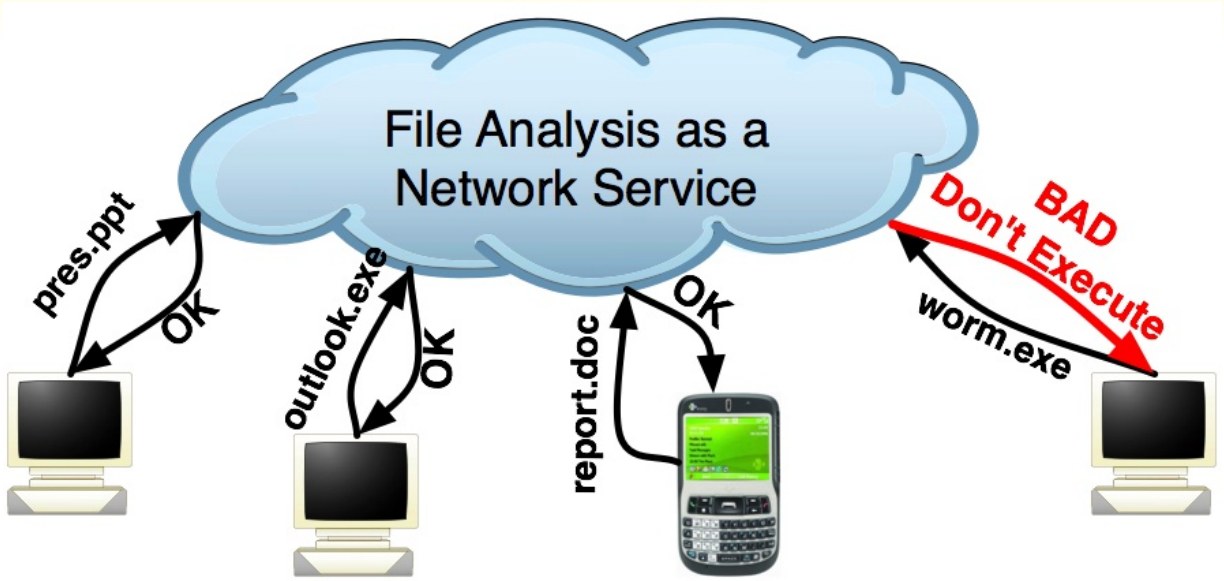
Roadmap

- Motivation and Limitations of Antivirus
- **AV as an In-Cloud Network Service**
- Implementation and Evaluation
- Discussion and Wrap-up

AV as a In-Cloud Network Service

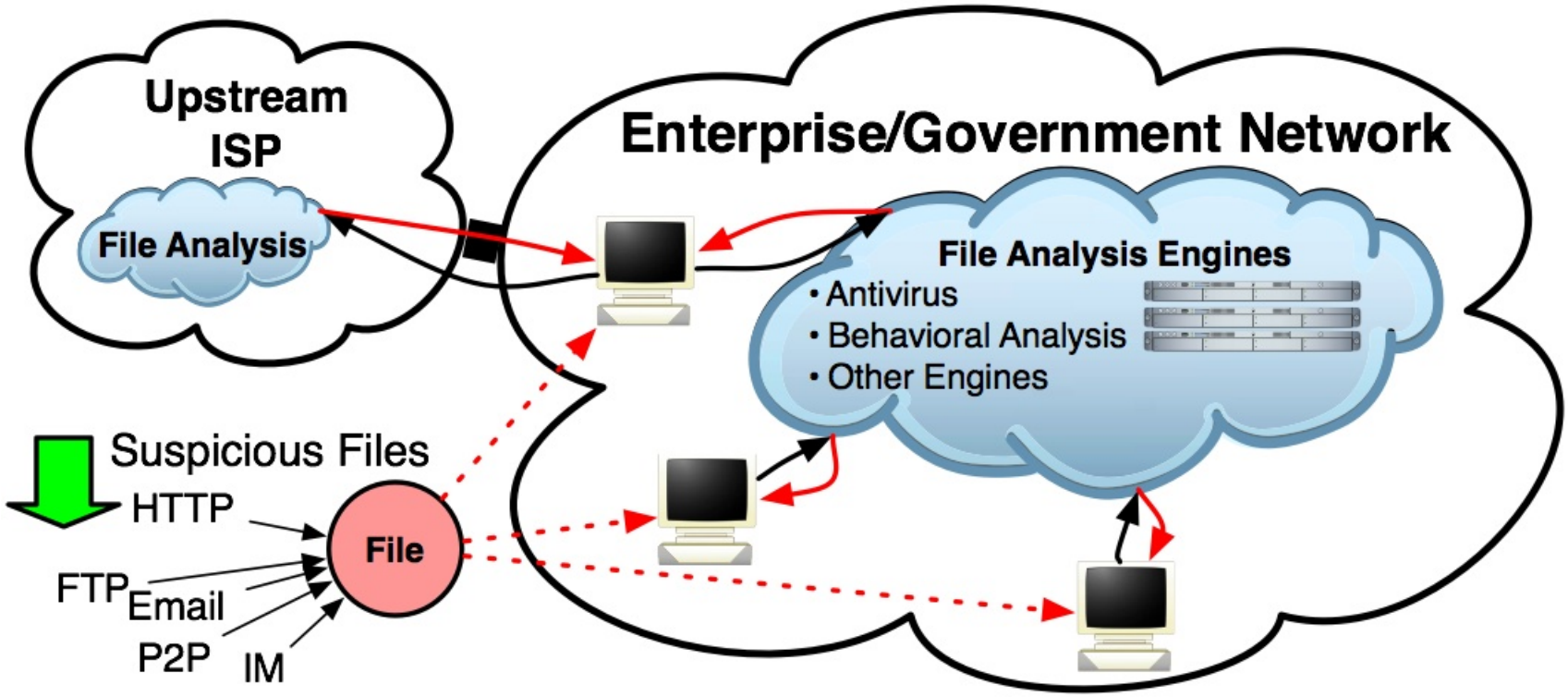
Network Component

Host Component



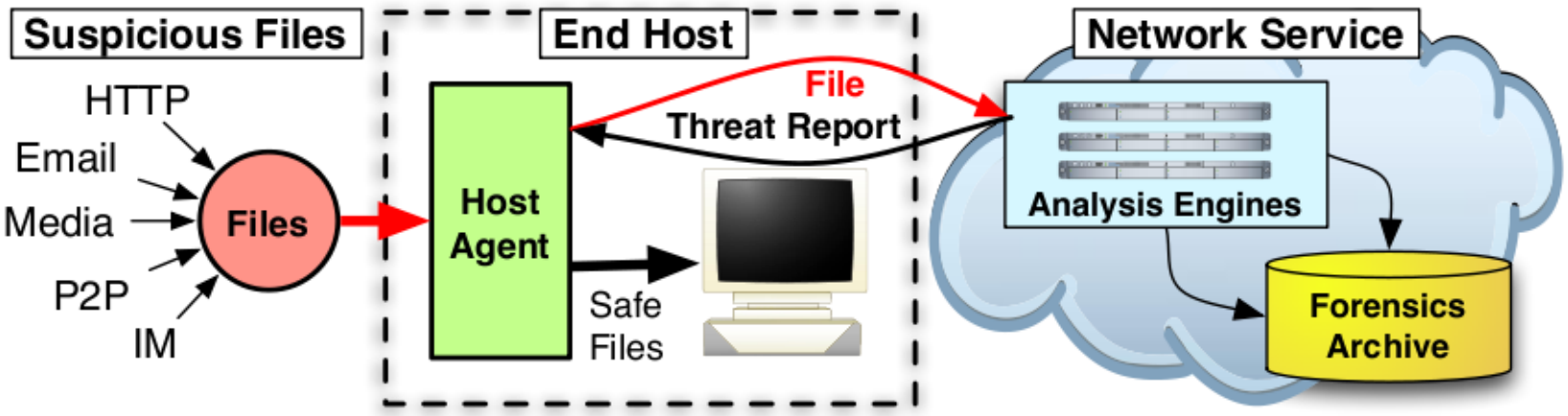
- By providing antivirus as an in-cloud service:
  - Analyze files using **multiple detection engines** in parallel
  - Collect **forensic data** for post-infection assessment
  - **Retrospectively detect** previously infected hosts
  - **Simplify host software** for wide deployability
  - Centralize **management** and policy enforcement

Deployment Model



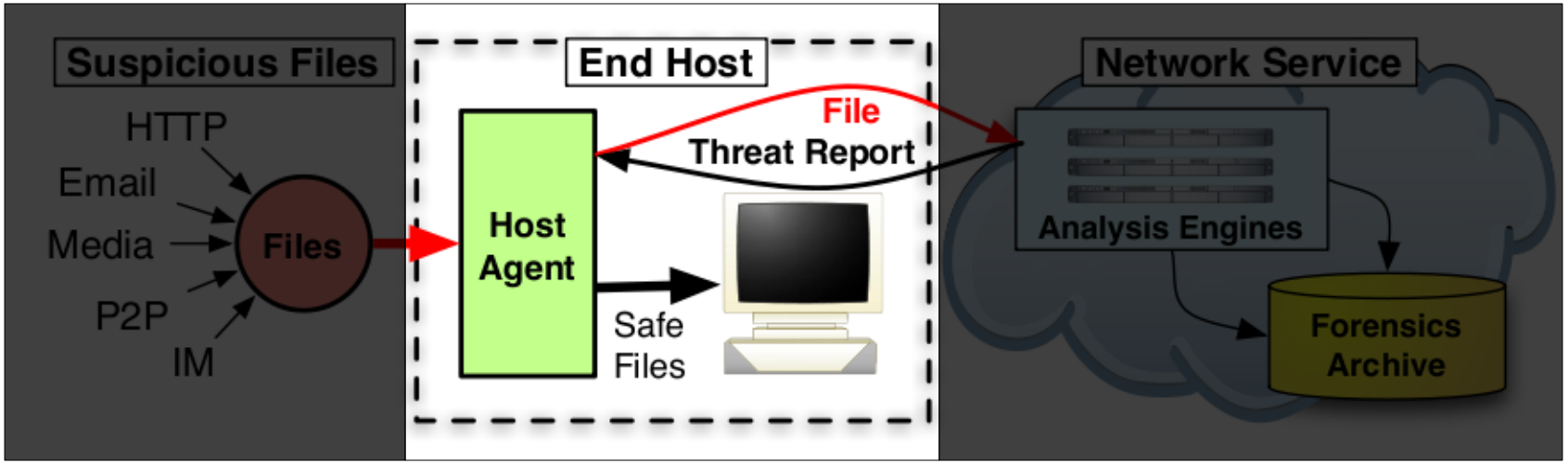
- Network service can be deployed inside an organization or by an upstream ISP

Architecture



- ***Lightweight host agent*** runs on desktops, laptops, and other devices
- ***Network service*** hosts the backend file analysis engines and fields requests from the host agent.
- ***Archival and forensics service*** stores information on file analysis results and provides a query and alerting interface

Architecture



- ***Lightweight host agent:***
  - Access to each file is trapped and diverted to a handling routing
  - Generate a unique identifier for the file (eg. cryptographic hash)
  - Compare UID to local and remote cache of previously analyzed files; send file to network service if not in either cache

Simplified Host Agent

Key insight: separate acquisition of files from detection routines; move complexity off end host

- Small code base → reduced vulnerability footprint
- Isolation from vulnerabilities present in the detection engines
- Easier to port to new operating systems

Simplified Host Agent

Cross Platform

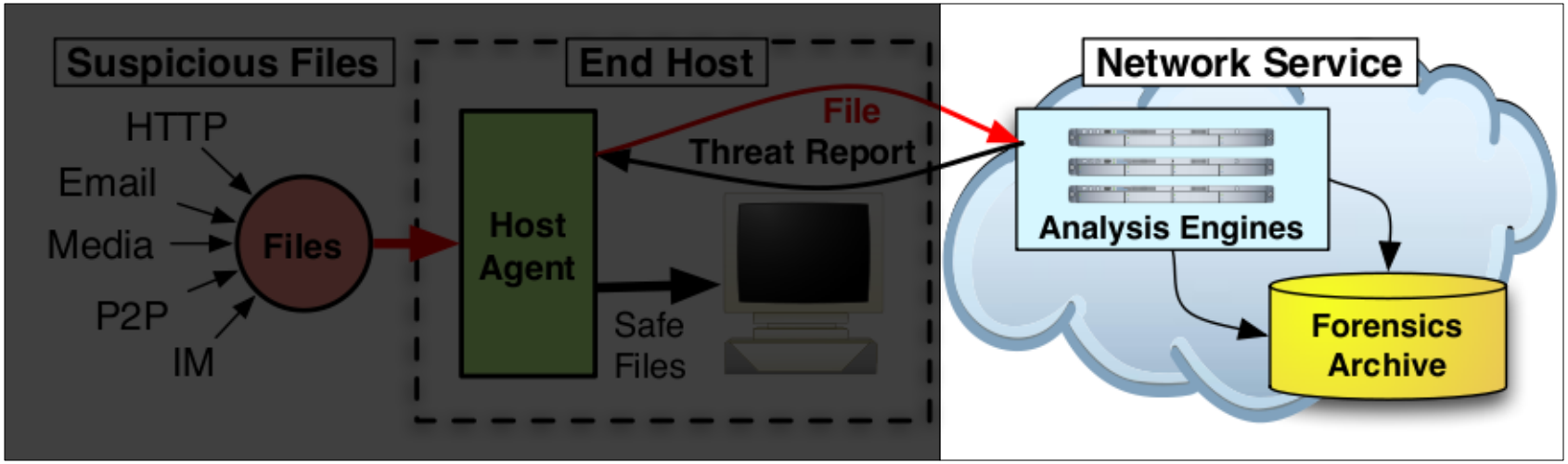



Mobile Devices

Mail Server Frontends




Architecture

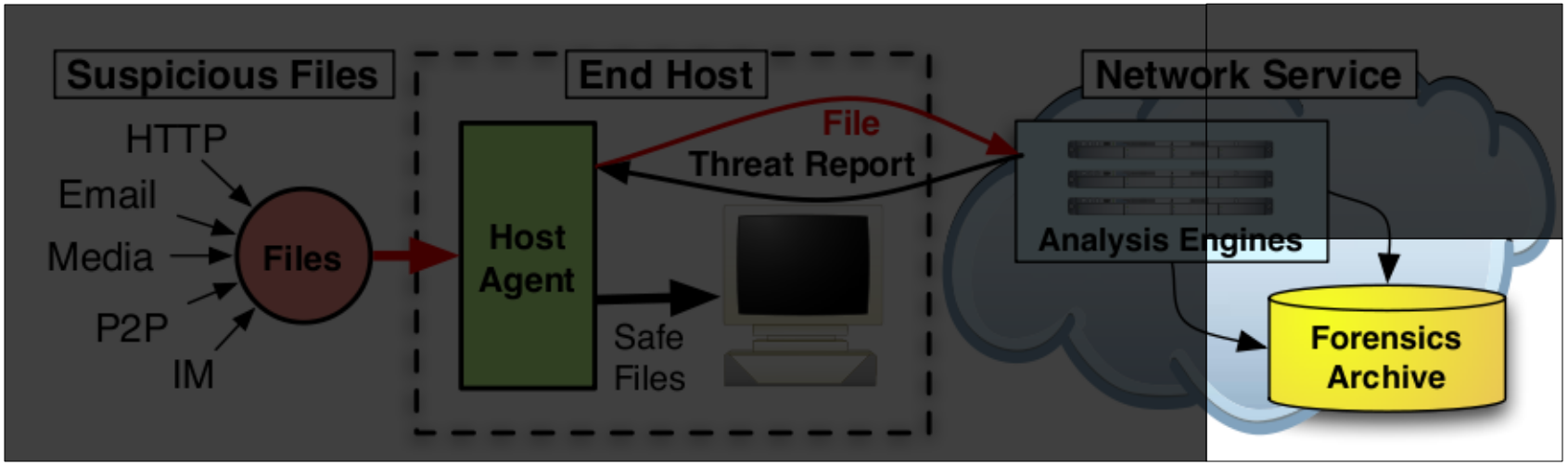


- ***Network service:***
  - Receives incoming analysis requests from host agent
  - File analyzed by collection of engines (N-version protection)
  - Central management of signatures updates and security policies
  - Shared remote cache maintained in network service

N-Version Protection

- N-version programming
  - Multiple, independent implementations for robustness and reliability
  - Observation: independent implementations are unlikely to suffer same failures/bugs
- N-version protection
  - Multiple, independent implementations for the detection of malware
  - Observation: independent vendors have heterogeneous detection routines, malware collection methodologies, and response times
  - Leverage heterogeneity to increase coverage

Architecture

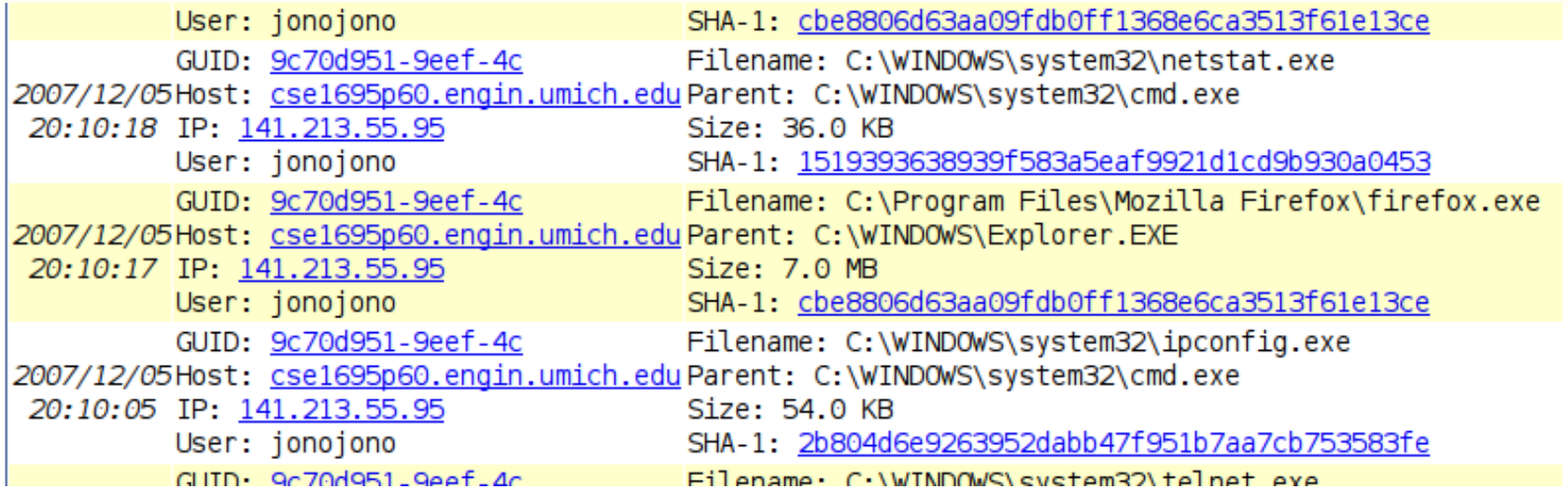


- ***Archival and Forensics Service:***
  - Retrospective detection: rescanning of archived files after a signature update; allows detection of previously infected hosts
  - Network-wide policy enforcement (for example: block unwanted applications, prevent execution of an email attachment)
  - Forensics tracking of file access

Retrospective Detection

- Detect previously unknown threats
- Host-based scenario:
  - Host infected by 0-day threat, antivirus disabled
  - Later: vendor releases new signatures to address threat
  - Result: sig updates not received, host infected indefinitely
- Network service with RD:
  - Host sends 0-day to network service, 0-day evades all detection engines, 0-day archived, host becomes infected.
  - Later: vendor releases new signatures to address threat. Network service rescans archived files, detects threat!
  - Result: Administrator notified of infected host, can quarantine, analyze forensics/behavioral information, disinfect.

Forensics Archive



	User: jonojono	SHA-1: cbe8806d63aa09fdb0ff1368e6ca3513f61e13ce
2007/12/05 20:10:18	GUID: 9c70d951-9eef-4c Host: cse1695p60.engin.umich.edu IP: 141.213.55.95 User: jonojono	Filename: C:\WINDOWS\system32\netstat.exe Parent: C:\WINDOWS\system32\cmd.exe Size: 36.0 KB SHA-1: 1519393638939f583a5eaf9921d1cd9b930a0453
2007/12/05 20:10:17	GUID: 9c70d951-9eef-4c Host: cse1695p60.engin.umich.edu IP: 141.213.55.95 User: jonojono	Filename: C:\Program Files\Mozilla Firefox\firefox.exe Parent: C:\WINDOWS\Explorer.EXE Size: 7.0 MB SHA-1: cbe8806d63aa09fdb0ff1368e6ca3513f61e13ce
2007/12/05 20:10:05	GUID: 9c70d951-9eef-4c Host: cse1695p60.engin.umich.edu IP: 141.213.55.95 User: jonojono	Filename: C:\WINDOWS\system32\ipconfig.exe Parent: C:\WINDOWS\system32\cmd.exe Size: 54.0 KB SHA-1: 2b804d6e9263952dabb47f951b7aa7cb753583fe
	GUID: 9c70d951-9eef-4c	Filename: C:\WINDOWS\system32\telnet.exe

- Contextual file access info
  - Temporal and causal relations between events
  - Drill down to *who/what/where/when* of infection
- Detailed runtime behavioral profiles
  - Enhanced *what*: feedback from behavioral engines
  - Assists in post-infection cleanup and risk assessment

Roadmap

- Motivation and Limitations of Antivirus
- AV as an In-Cloud Network Service
- **Implementation and Evaluation**
- Discussion and Wrap-up

Implementation – Host Agent

- Platforms:
  - Windows 2000/XP/Vista, Linux 2.4/2.6, FreeBSD 6
  - Milter frontend interface (Sendmail, Postfix)
  - Nokia Maemo mobile platform
- Win32 host agent
  - Win32 API hooking (jmp insertion, IAT/EAT patching)
  - ~1500 LOC, 60% managed code
  - Co-exists peacefully with existing AV engines
- Linux/BSD host agent
  - Python, < 300 LOC, LSM syscall hooking

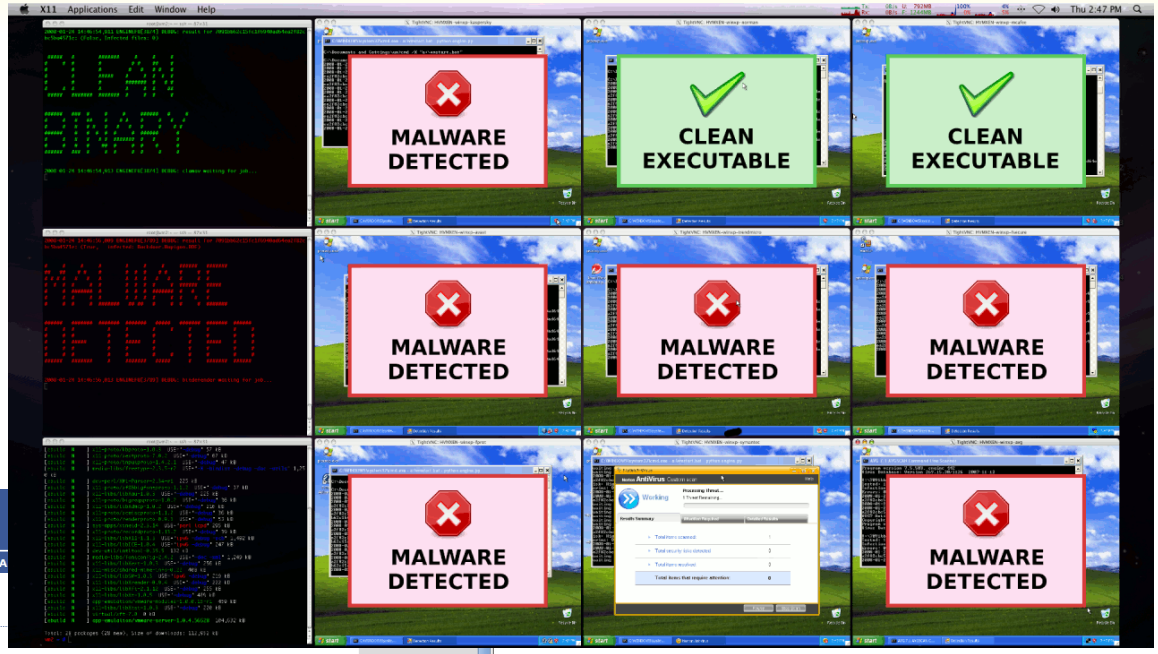
Implementation – Network Service

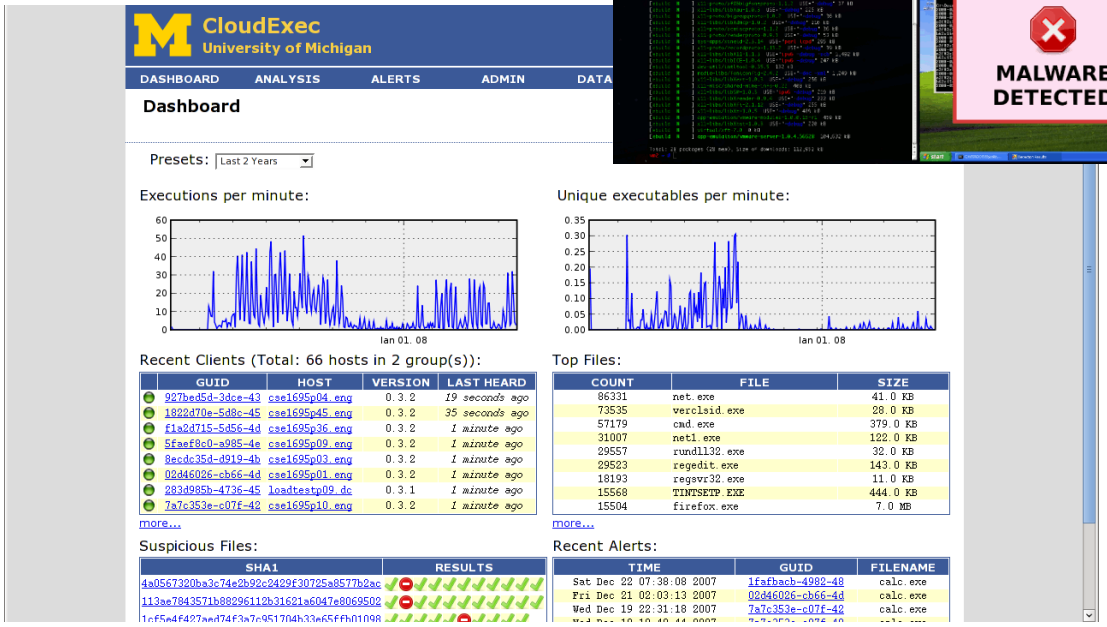
- Backend analysis engines
  - 10 antivirus engines:
    - Avast, AVG, BitDefender, ClamAV, F-Prot, F-Secure, Kaspersky, McAfee, Symantec, Trend Micro
  - 2 behavioral engines
    - Norman Sandbox, CWSandbox
- Hosted in Xen VM containers
  - 9 WinXP HVM, 3 Linux domU paravirt
  - Isolation/Recovery: in case of engine compromise
  - Scalability: dynamically spin up/down instances

Management Interfaces

Web interface:

- Forensics Drilldown
- Policy Enforcement
- Flexible Alerting
- Report Generation





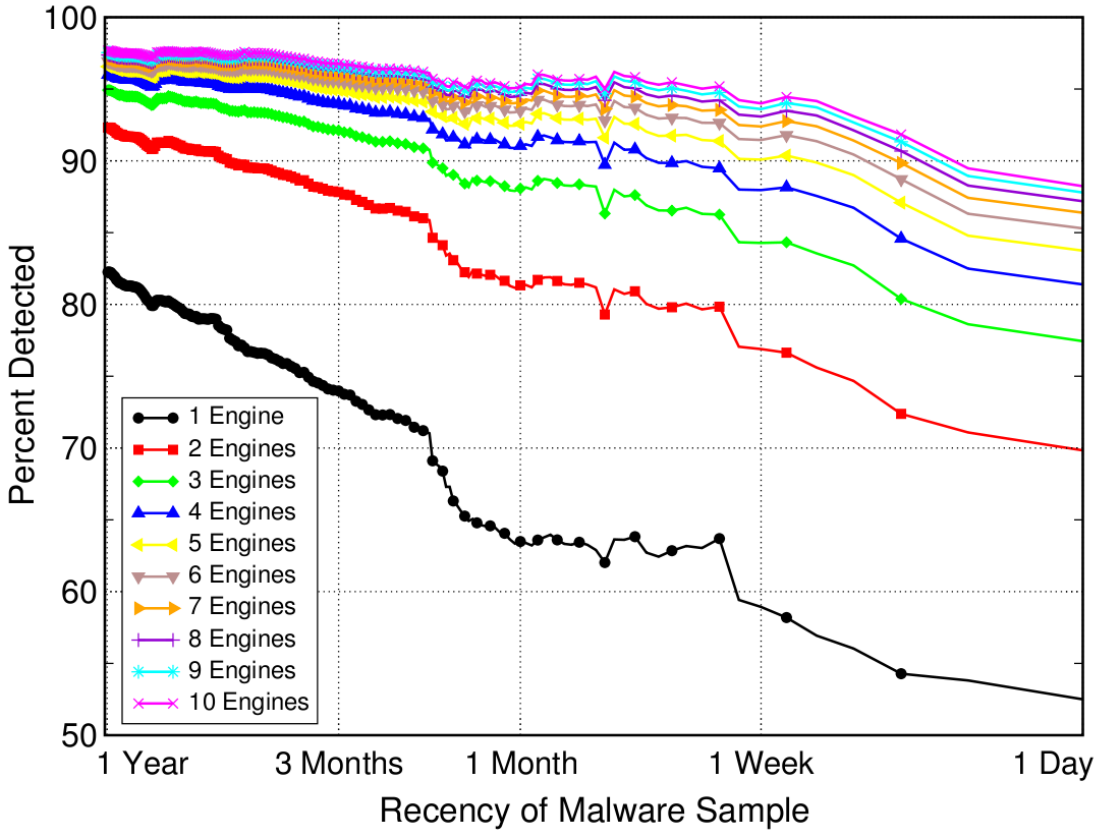
VM Monitoring:

- Real-time System Status
- Xen VM Management
- Visualization Eye-Candy!

Evaluation

- Malware Dataset
  - Arbor Malware Library (AML)
  - 7220 malware samples
  - Collected over a year period
- Deployment Results
  - Production deployment on campus network
  - Win32 host agent in computing labs
  - Over 6 months of data

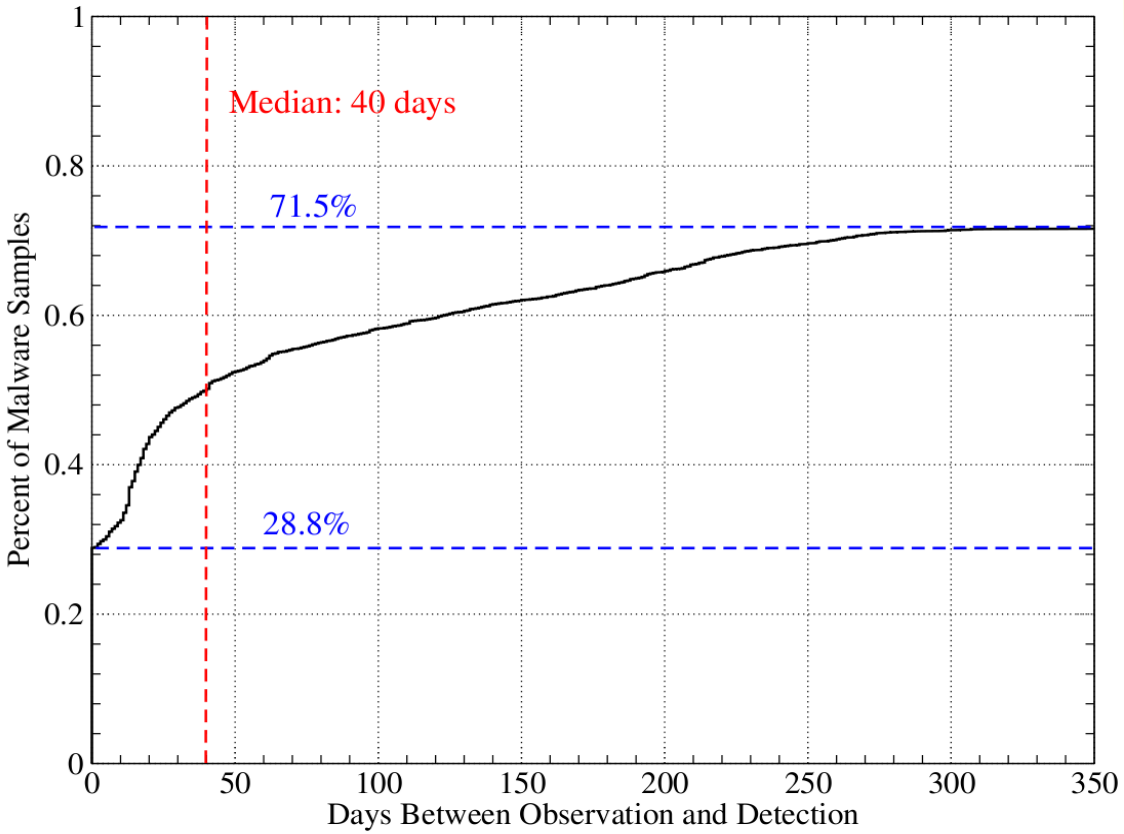
N-Version Protection



- **Single engine from 82% to 52%**
- **Ten engines from 98% to 88%**
- **For zero-day 88% vs. 52%**
- **Diminishing marginal utility**

Detection rates are calculated by taking the average rate across all combinations of N engines.

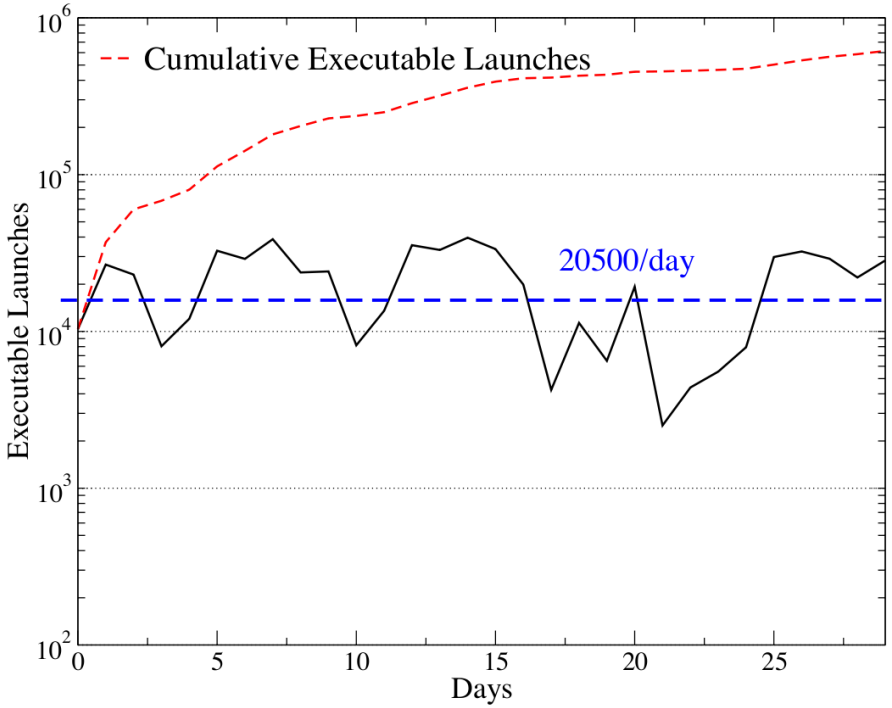
Vulnerability Window



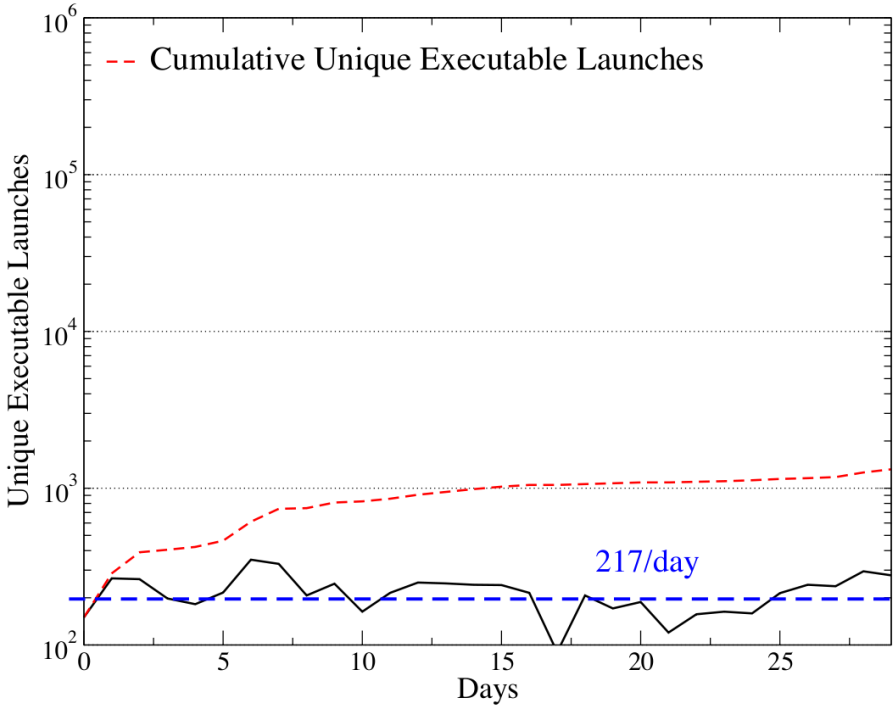
- AML data set + archive of McAfee DAT signature files with 1-week granularity
- ~30% samples already detected
- ~30% samples never detected
- ~70% eventually detected
- Average time between observation and detection is 48 days.

Large window of vulnerability: retrospective detection is essential to discover previously infected hosts

Caching and Performance



615K execution events



1300 unique executables

99.8% remote cache hit rate: files rarely need to be transferred to network service for analysis

Bandwidth and Latency

- Boot Process: 10 processes
  - Warm local: none
  - Warm remote: 8.7 kb
- Login process: 52 processes
  - Warm local: none
  - Warm remote: 46.2 kb
- Comparison: Active Directory (LDAP)
  - Boot: 171 kb
  - Login: 270 kb
- Average binary analysis time:
  - 1.3 seconds

Roadmap

- Motivation and Limitations of Antivirus
- AV as an In-Cloud Network Service
- Implementation and Evaluation
- **Discussion and Wrap-up**

Discussion

- Disconnected operation
  - Local caching, policy decision
- False positives
  - Confidence, engine thresholds, management
- Detection engine licensing
  - Price/perf, free engines, lock-in
- Sources of malicious code
  - DLL results, file types configurable
- Context and environment
  - Can execute candidate files in VM
- Privacy implications
  - Tunable collection and display

Threshold	False Positives	Detection
1	12	97.7%
2	5	96.3%
3	2	95.2%
4	0	93.9%

AV Vendor	1 Week
Avast	+24.6%
AVG	+8.7%
BitDefender	+3.1%
ClamAV	+0.0%
F-Prot	+12.6%
F-Secure	+15.0%
Kaspersky	+2.3%
McAfee	+14.2%
Symantec	+20.6%
Trend Micro	+12.6%

Wrap-up

An in-cloud service is an intuitive and effective approach for malware detection.

Adhoc solution → In-Cloud solution

- In-Cloud advantages
  - Global visibility (inherent)
  - Centralized management (inherent)
  - Application-specific advantages

- Past in-cloud services
  - Email filtering
  - DDoS mitigation
  - Phishing

- Future in-cloud services
  - HIDS
  - Anomaly detection
  - ???

Question and Answer

Questions?

- Contact information
  - Jon Oberheide
  - University of Michigan
  - jonojono@umich.edu
  - http://www.eecs.umich.edu/fjgroup/