Virtualized In-Cloud Security Services for Mobile Devices

Jon Oberheide, Kaushik Veeraraghavan, Evan Cooke, Jason Flinn, Farnam Jahanian

University of Michigan

June 17, 2008

MobiVirt '08

Roadmap

- Motivation
  - Malware detection: high security, low resources
- Architecture
  - Antivirus as an off-device network service
- Implementation and Evaluation
  - Resource requirements
  - Power consumption
  - Detection capabilities
- Discussion and Wrap-up

Motivation

- Mobile device capabilities increasing
  - Approaching functionality of standard PCs
  - Rich application development/delivery
  - iPhone, Android, Maemo, Symbian, WM, etc
- Enticing target for malware authors
  - Mobile banking transactions
  - Spying on business/enterprise users
- Need malware detection/mitigation!

Current Approach

- Adapt host-based antivirus to mobile devices
  - Numerous vendors have mobile products
- Problems with on-device AV model
  - Detection capabilities vs. resource constraints
  - Scalability protection for future threats
  - Software complexity, platform diversity, AV vulnerabilities

Goal: maintain/increase detection capabilities while reducing resource requirements

New Approach

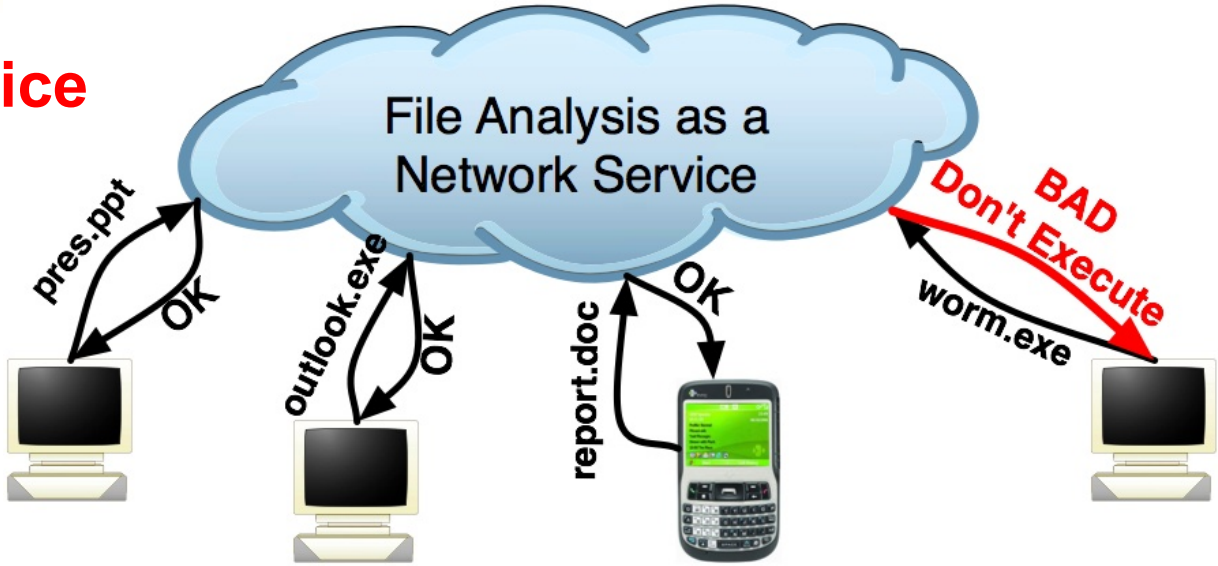
- Offload detection functionality
  - Instead of analyzing a file locally on-device, send it to a network service for analysis
- Moves complexity and resource usage to off-device service
  - Frees us of resource constraints of mobile device
- Trade-off network bandwidth/radio power to save on-device resources

CloudAV Architecture

Network Service

Separating acquisition from analysis

Host Agent



- ***Host/mobile agent*** runs on desktops, laptops, and mobile devices.
  - Acts similar to host-based AV; interposes on file access
  - Queries network service instead of analyzing locally
- ***Network service*** hosts the backend AV detection engines and fields file analysis requests from the host agent.

Architecture Components

- Lightweight Mobile Agent
  - Low resource requirements
  - Easy to port to new platforms
  - Simple code base → reduced vuln footprint
- Network Service
  - Can employ multiple detection engine in parallel
  - Central management of AV signature updates
  - More resource intensive and complex detection techniques (eg. behavioral detection engines)

Advantages of Virtualization

- Network service backend
  - Hosts detection engines in virtualized environments
- Scalability
  - Dynamically spin up/down instances
- Isolation/Recovery
  - AV engine vulnerabilities
  - Restore to clean snapshot when compromise detected

Caching, Caching, Caching

- Reducing network activity is desirable
  - Transferring candidate file to network service for analysis on every access is infeasible
- Remote cache
  - Shared between all users of network service
  - Eliminate duplicate file transfer and analysis:
    - Alice runs App1, App1 analyzed; Bob runs App1, remote cache hit!
- Local cache
  - Stored on mobile device
  - Eliminate unnecessary remote cache queries
    - Bob runs App1, remote cache hit; Bob runs App1 again, local cache hit!

Implementation – CloudAV

- Host Agent
  - Numerous platforms: Win32, BSD, Milter frontends
- Network Service:
  - 10 antivirus engines:
    - Avast, AVG, BitDefender, ClamAV, F-Prot, F-Secure, Kaspersky, McAfee, Symantec, and Trend Micro
  - 2 behavioral engines
    - Norman Sandbox, CWSandbox

Implementation – Mobile Extensions

- Nokia Maemo platform
  - N770, N800, N810 devices
  - Python, < 300 LOC, Dazuko syscall hooking
- Mobile-specific behavioral engine
  - Runtime behavioral analysis of suspected malware
  - Virtualized Maemo environment
  - Monitors syscalls and D-Bus IPC to detect:
    - Modification/destruction of personal user data
    - Network communications to untrusted parties
    - Skype VoIP calls to unauthorized numbers

Evaluation

- Resource macro-benchmark
  - Nokia N800
  - ClamAV vs. Mobile Agent (MA)
  - Resource consumption (CPU/memory)
- Power consumption micro-benchmark
  - Nokia N95
  - Kaspersky Mobile vs. Mobile Agent (MA)
  - Power consumption of radios (WiFi/EDGE)
- Mobile agent cache states
  - CL+CR, CL+WR, WL+WR

Computational Resources

- Simulated real-world usage benchmark
  - 5 common applications: web browser, IM client, VoIP client, media player, and PDF viewer

Agent	Startup Time	Avg/Peak Memory	User/Total Jiffies
ClamAV	57 sec	25967 KB / 39556 KB	13349 / 15684
MA-CL+CR	0.2 sec	1502 KB / 2154 KB	1502 / 2185
MA-CL+WR	0.2 sec	1486 KB / 2124 KB	1486 / 1854
MA-WL+WR	0.2 sec	1189 KB / 1812 KB	1189 / 1714

Order of magnitude decrease in CPU and memory resources in all mobile agent cache states.

Power Consumption

- Local Kaspersky vs. Mobile Agent
  - Simple scan of ~25M of third-party apps and games

Agent	Avg Energy	Peak Energy	Total Energy
None (Baseline)	0.36 W	0.63 W	43.2 W
Kaspersky	0.86 W	1.27 W	89.4W
MA-CL+CR (EDGE)	1.51 W	2.31 W	250.6 W
MA-CL+CR (WiFi)	1.31 W	2.44 W	165.1 W
MA-CL+WR (EDGE)	1.22 W	2.13 W	126.9 W
MA-CL+WR (WiFi)	0.92 W	1.83 W	74.5 W
MA-WL+WR	0.82 W	1.20 W	59.5 W

Decrease in power consumption in common mobile agent cache states.

Scalability of Detection Capabilities

- Current state of mobile device malware
  - Presently not a large number of threats to protect against
- Host-based AV model
  - Resource/power consumption of on-device software scales with complexity/number of threats

Detection Engine	Signature Database Size
Symantec Mobile	27 signatures
Kaspersky Mobile	284 signatures
ClamAV	262,289 signatures
Cloud AV / Mobile Agent	> 5 million sigs + behavioral

- CloudAV model
  - Resource/power consumption stays static and independent of increase in threats

Detection Coverage

- N-Version Protection
  - The use of multiple detection engines in parallel from independent vendors
- Desktop malware results
  - Subset of malware from Arbor Malware Library

Engine Combination	Detected	Coverage
CM	229/469	48.82%
CM, SM	290/469	61.83%
CM, SM, MA	358/469	76.33%
CM, SM, MA, BD	417/469	88.91%
CM, SM, MA, BD, FS	430/469	91.68%

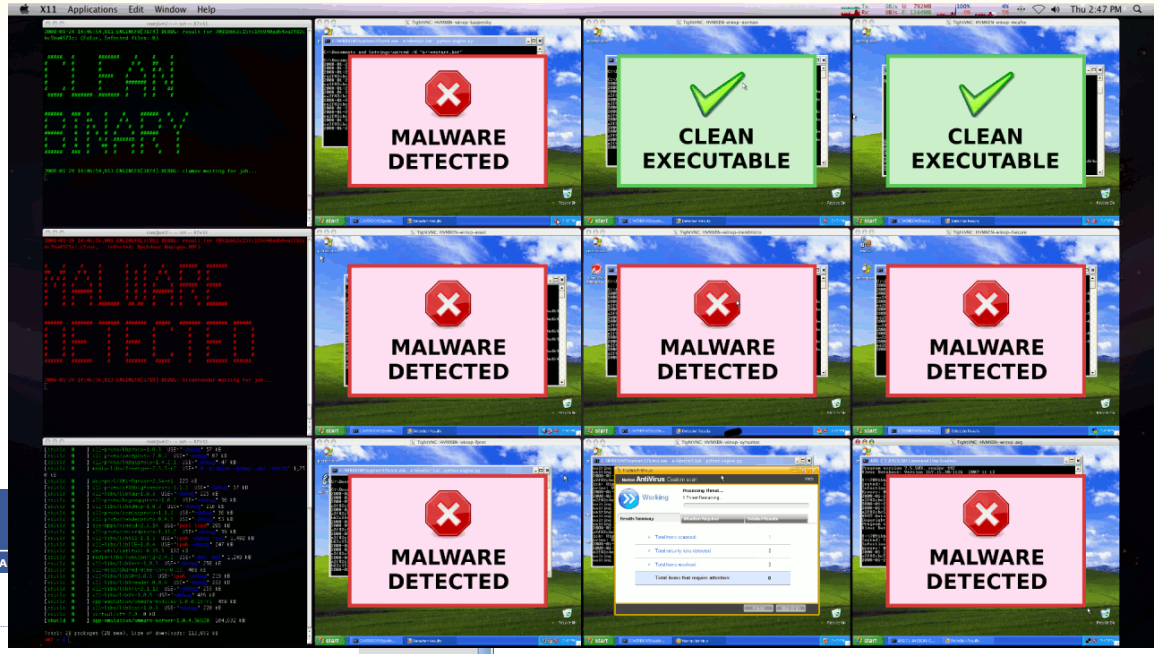
CM: ClamAV
SM: Symantec
MA: McAfee
BD: BitDefender
FS: F-Secure

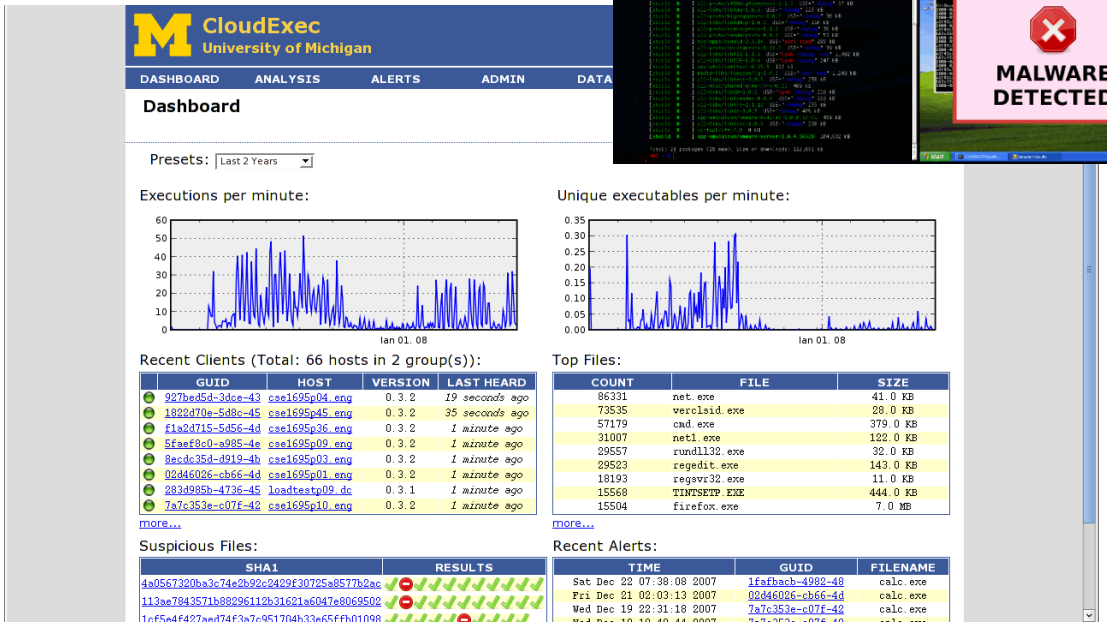
- Transparent engine extensibility

Management Capabilities

Web interface:

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





VM Monitoring:

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

Limitations

- Disconnected operation
  - Local caching mechanisms
  - Limited app/content acquisition while disconnected
  - Security policy decision
- Privacy concerns
  - Tunable collection/display built into architecture
  - User awareness of operation

Question and Answer

Questions?

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