Automated Identification of Installed Malicious Android Applications

By

Mark Guido, Justin Grover, Jared Ondricek,
Dave Wilburn, Drew Hunt and Thanh Nguyen

Presented At

The Digital Forensic Research Conference

DFRWS 2013 USA Monterey, CA (Aug 4th - 7th)

Detecting Maliciousness Using Periodic Mobile Forensics

Authors: Mark Guido, Jared Ondricek, Justin Grover,
David Wilburn, Thanh Nguyen, Drew Hunt

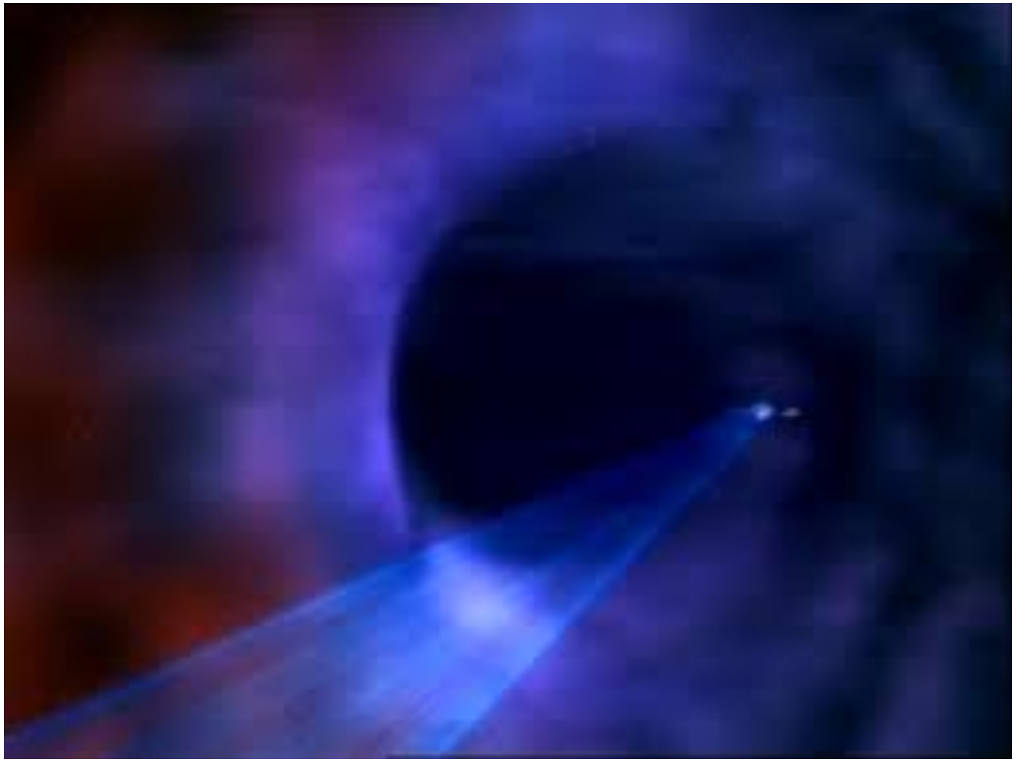
Approved for Public Release: 12-4346. Distribution Unlimited.

Problem

- Use case
  - Enterprise deployments
- Be proactive!
  - Can we apply media forensics techniques to detect activities that are indicative of malicious behavior?
  - Measure changes to block devices
- Organizations no longer own their phone infrastructure
  - Centrally manage and audit phone usage

We want to make use of a richer set of user data on a phone as compared to a laptop

Approved for Public Release: 12-4346. Distribution Unlimited.

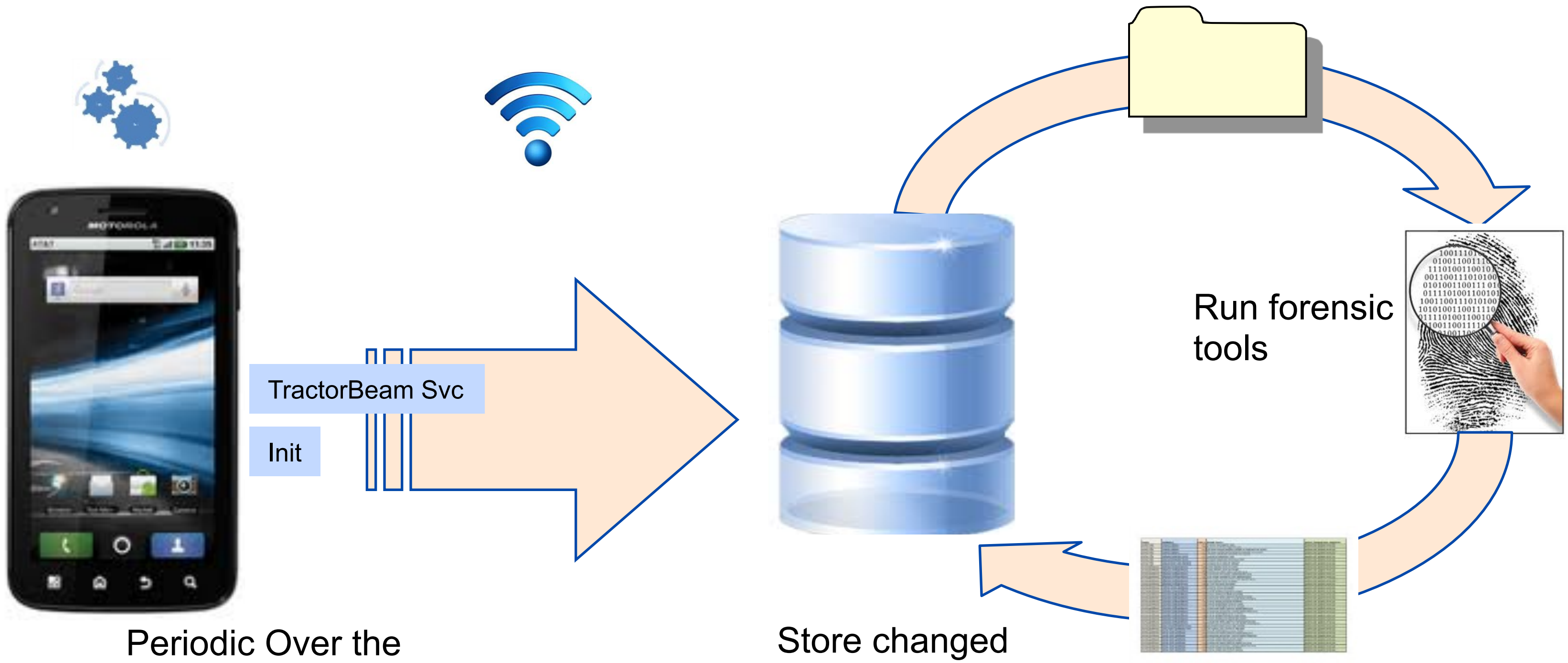


Client, Server, Database, Analysis Framework, Forensic Tools

Tractor Beam

Process – Live Phone Forensics

Reassemble forensic images



Run forensic tools

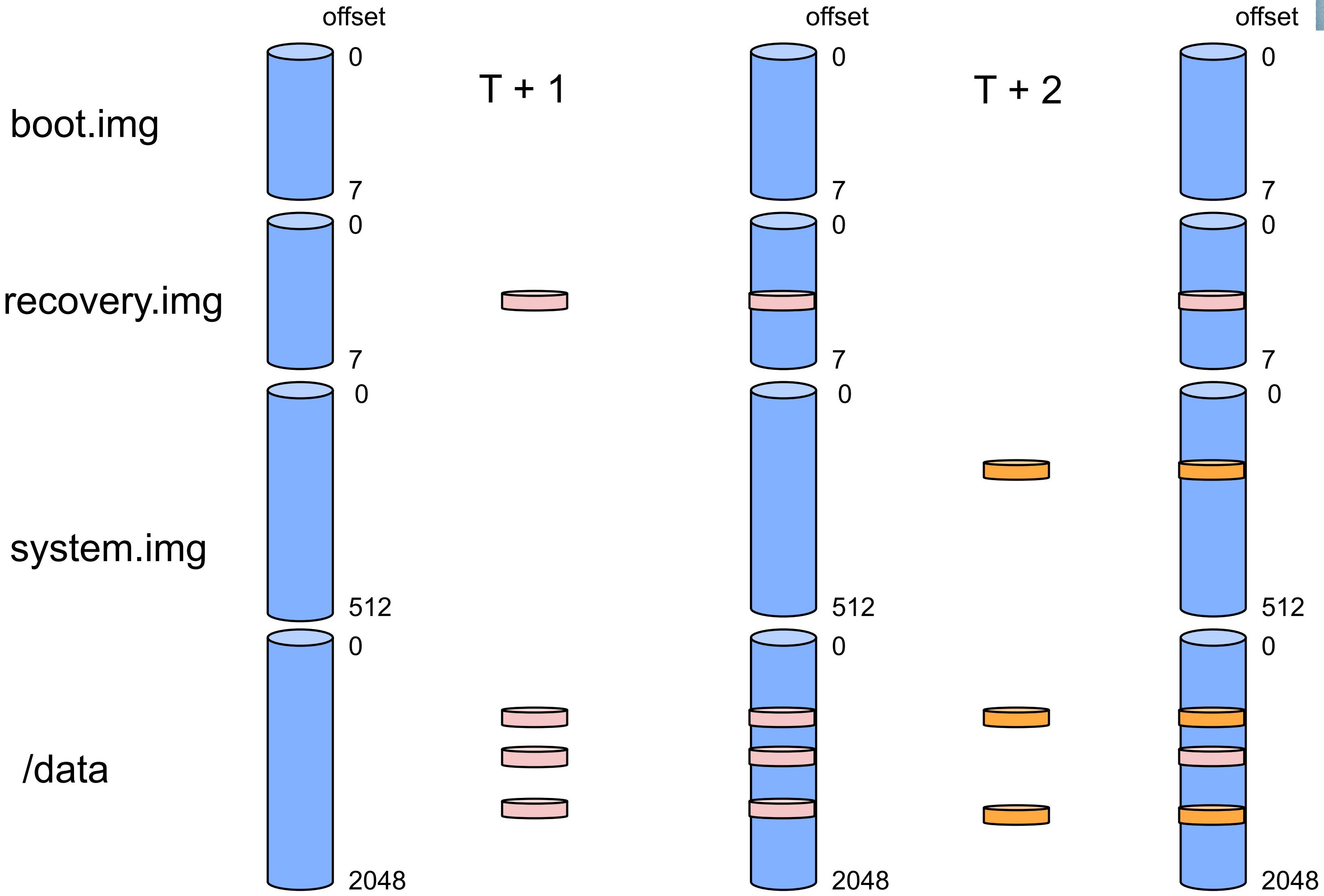
Periodic Over the Air (OTA) transmission to server

Store changed data

Extract audit information

All forensic techniques are performed offline on image snapshots

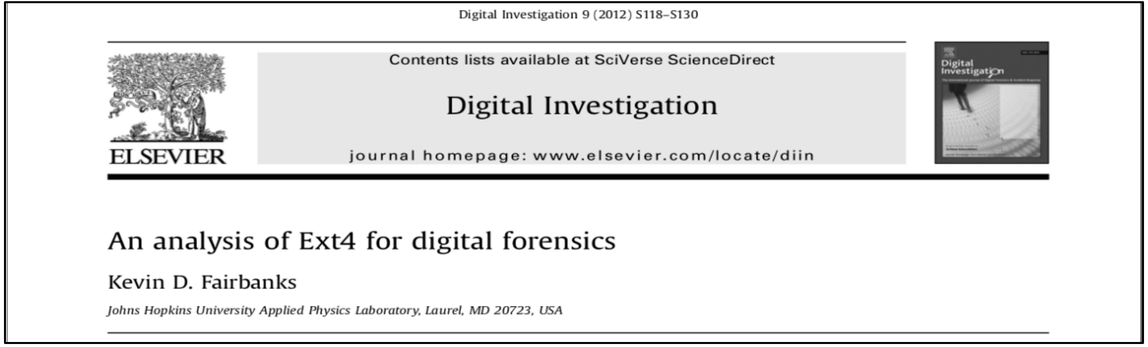
Reconstructing Images



Approved for Public Release: 12-4346. Distribution Unlimited.

Forensic Tools

- Modern phones run Fourth Extended Filesystem (ext4)
  - Can apply forensic tools on reconstructed images
    - The Sleuth Kit 4.0.0b1 with experimental ext4 support
      - Identify when .apks are installed
      - Identify deleted files
      - Reconstruct .apk file
    - Fiwalk 0.6.15
      - Identify all file system Modified, Accessed, Created, Entry Modified (MACE) times
      - Generate DFXML to feed analysis tools



Digital Investigation 9 (2012) S118–S130

Contents lists available at SciVerse ScienceDirect

Digital Investigation

journal homepage: www.elsevier.com/locate/diin

ELSEVIER

An analysis of Ext4 for digital forensics

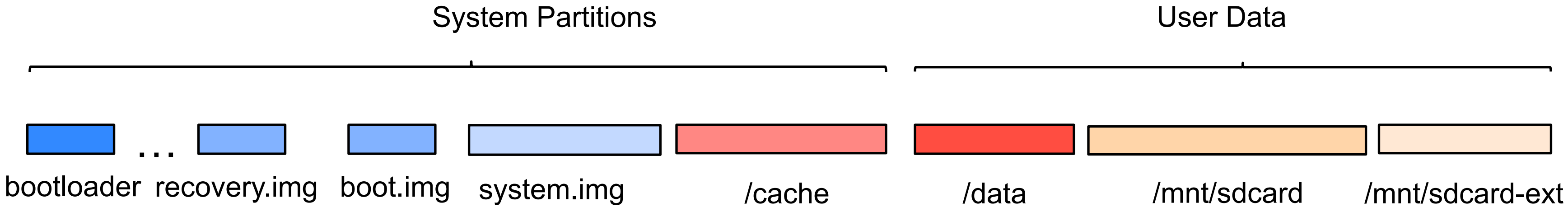
Kevin D. Fairbanks

Johns Hopkins University Applied Physics Laboratory, Laurel, MD 20723, USA

Approved for Public Release: 12-4346. Distribution Unlimited.

Mobile Malware

- Certain phone partitions change continuously, others only when user initiates.



- Enterprise use case:
  - We should only rarely see changes to blue system partitions

Tractor Beam is already set up to easily identify when these changes occur

7 Detectors and Loggers

- Detector 1. Alerts on changes to boot.img
- Detector 2. Alerts on changes to recovery.img
- Detector 3. Alerts on changes to bootloader
- Detector 4. Alerts on changes to system.img
  - Useful for establishing persistence, surviving a reboot
- Logger 5. Compares image, logs all timestamp MACE time changes since previous snapshot
- Logger 6. Identifies and logs all deleted files since previous snapshot
- Detector 7. Identifies newly installed .apks and parses AndroidManifest.xml for BOOT_COMPLETED registration

Approved for Public Release: 12-4346. Distribution Unlimited.

Experimentation

- Needed real mobile malware
- Android Malware Genome Project
  - MITRE obtained dataset of 1200 samples
  - No source code
    - Functionality of samples were not fully characterized
- Ran experiments in MITRE’s Network Attack Investigations Laboratory (NAIL)
  - No wireless, Faraday bag
  - No SIM card in phone
- Used real phone
  - Nexus S running Android 2.3.1

Approved for Public Release: 12-4346. Distribution Unlimited.

Experimentation

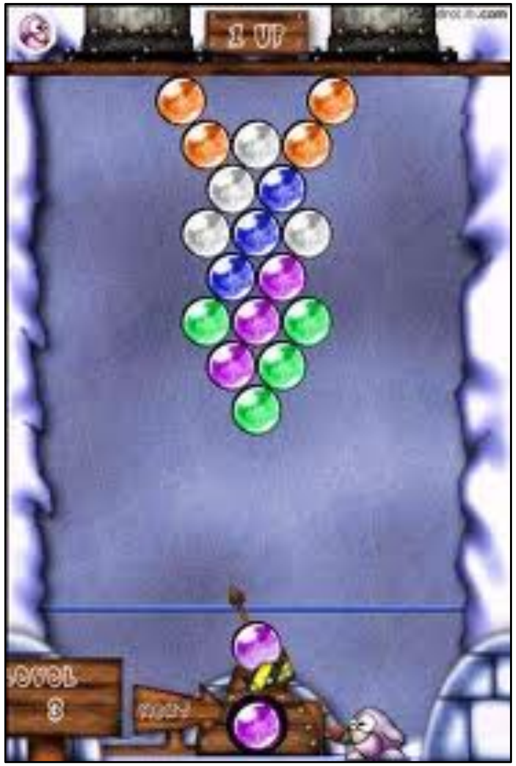
- **Round 1**
  - 20 malware apps
  - Goal: Test effectiveness of detectors

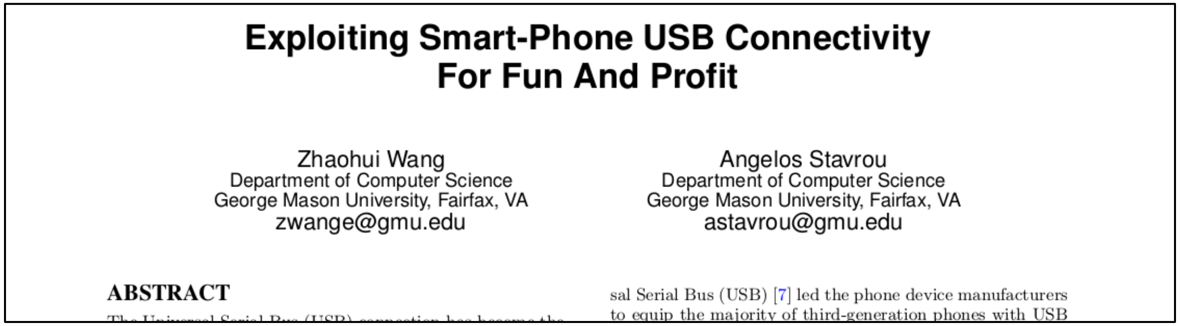
- **Round 2**
  - 100 apps
    - 90 legitimate
    - 10 malware
  - Goal: Test the “BOOT_COMPLETED” detector

- **Round 3**
  - Custom malware app
  - Goal: Test effectiveness against sophisticated malware

Round 3 – Frozen Bubble

- Demo malware
  - Have source code
- Targeted at Nexus S
  - Gingerbreak exploit
- Contains malware-like capabilities
  - Modifies boot.img for persistence
  - Modifies system.img - root
  - Phone emulates USB keyboard





Exploiting Smart-Phone USB Connectivity For Fun And Profit

Zhaohui Wang
Department of Computer Science
George Mason University, Fairfax, VA
zwange@gmu.edu

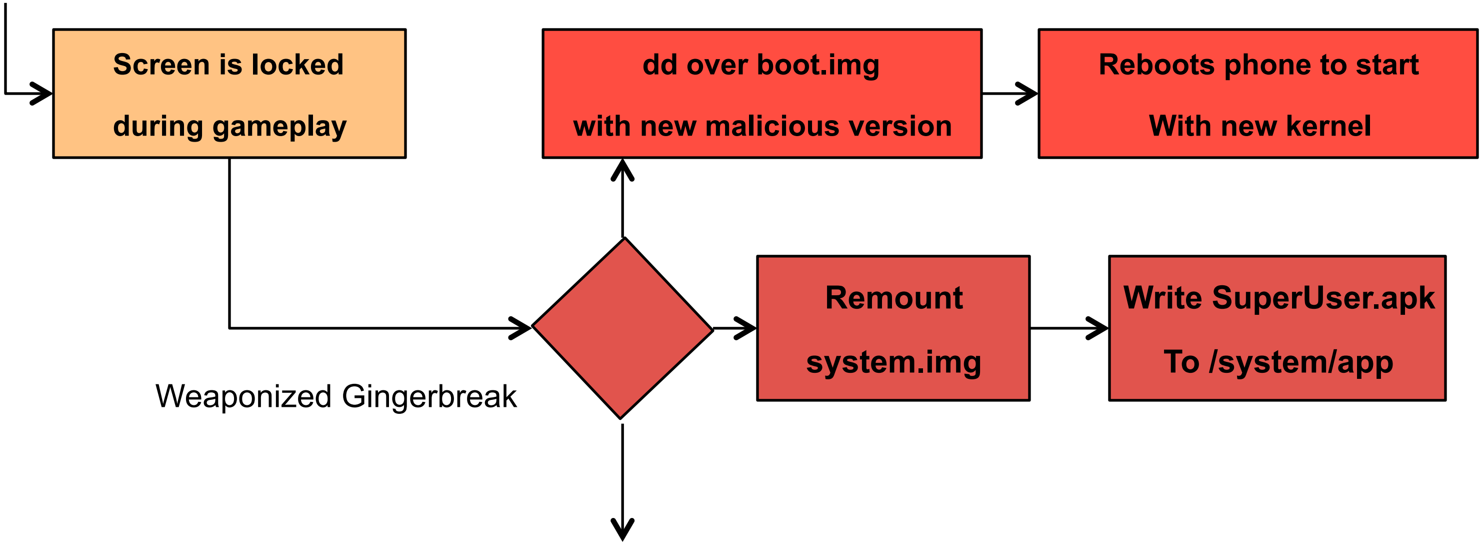
Angelos Stavrou
Department of Computer Science
George Mason University, Fairfax, VA
astavrou@gmu.edu

ABSTRACT

The Universal Serial Bus (USB) connection has become the

sal Serial Bus (USB) [7] led the phone device manufacturers to equip the majority of third-generation phones with USB

Round 3



Sample	Logged		Detected			Detection Result
	Installed	Exploit Dropped	BOOT_COMPLETED	system.img Change	boot.img Change	
1	x	x		x	x	Success

Application dropped no files and installed to /mnt/secure/asec.

Approved for Public Release: 12-4346. Distribution Unlimited.

Summary

- Project focused on mobile malware on the enterprise:
  - Paper concluded that there are alternative methods that showed promise for identifying and classifying mobile malware on running phone
- We also developed:
  - a method of extracting only changed blocks of data and reassembling an image from them
    - Normalized storage and fast reassembly
  - An analysis framework for running forensic detectors
    - 7 detectors were developed
- This provides us a platform for future work
  - New detectors are just Python scripts

Future Directions

- Insider threat – identifying events and patterns of events that are indicative of malicious behavior by the phone owners

- Masquerading users – identifying when phones may be being used by someone other than the phone owner based upon observed behavior
  - CERIAS collaboration
  - 30 Samsung Galaxy SIII’s

- Application of techniques for generalized forensics acquisition
  - Forensics laboratory use case

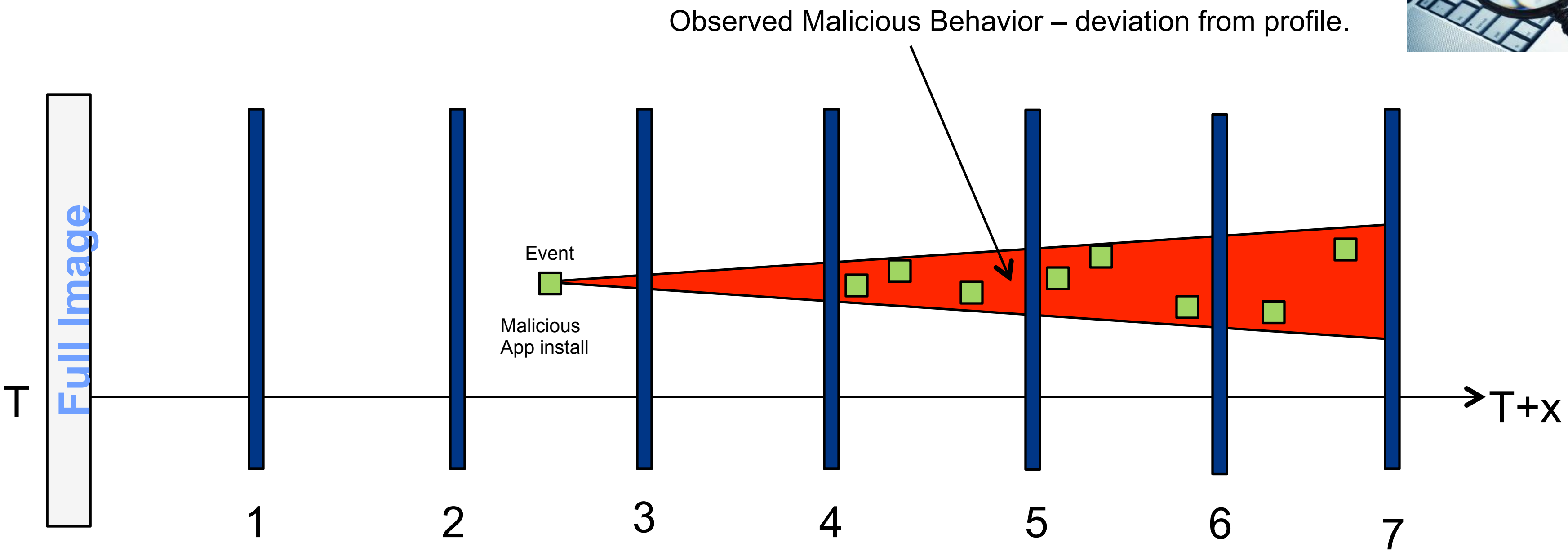
Approved for Public Release: 12-4346. Distribution Unlimited.

Questions?

Approved for Public Release: 12-4346. Distribution Unlimited.

Backup Slides

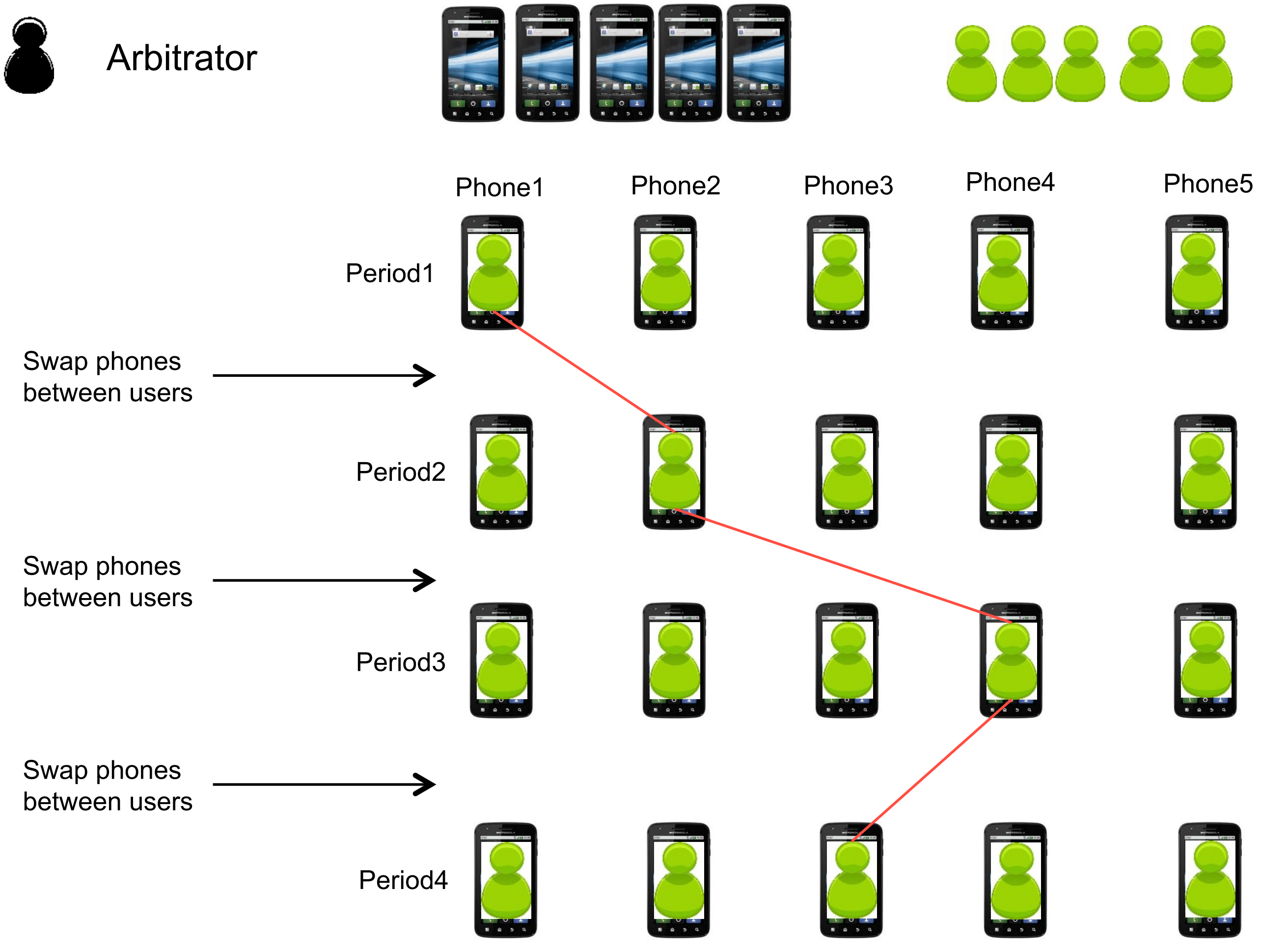
Approach



- Take initial full forensic image
- Periodically send only changed data Over The Air (OTA) to server
- Reconstruct images at collection times
- Run series of detectors that incorporate various best practice media forensic techniques
- Eventize the results

Approved for Public Release: 12-4346. Distribution Unlimited.

Masquerading Experimentation

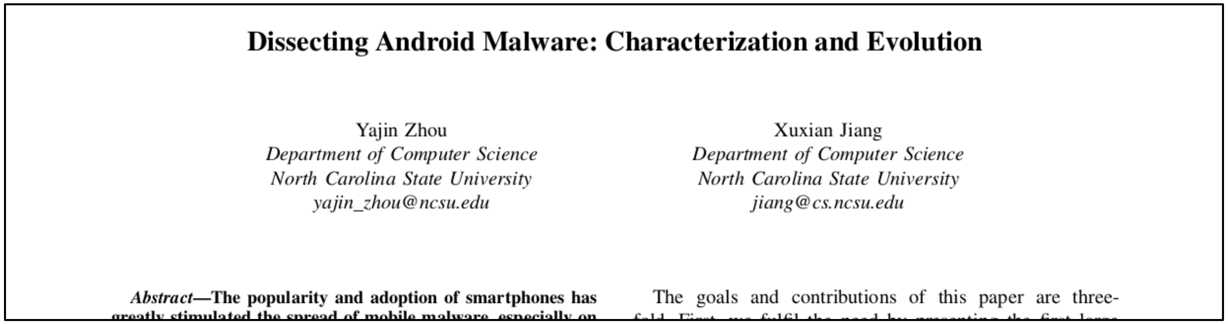

Arbitrator
Phone1
Phone2
Phone3
Phone4
Phone5
Period1
Swap phones between users
Period2
Swap phones between users
Period3
Swap phones between users
Period4

A successful result would be to identify each user/phone combination based solely on behavioral usage information.

Approved for Public Release: 12-4346. Distribution Unlimited.

BOOT_COMPLETED

- Applications can register to receive BOOT_COMPLETED event
  - Triggered when the phone finishes its boot process
  - Can use event notification to restart service
- 83% of samples in Android Malware Genome Project set registered for this event
- Must register to receive this event in the AndroidManifest.xml file
- .apk files typically installed in /data/app



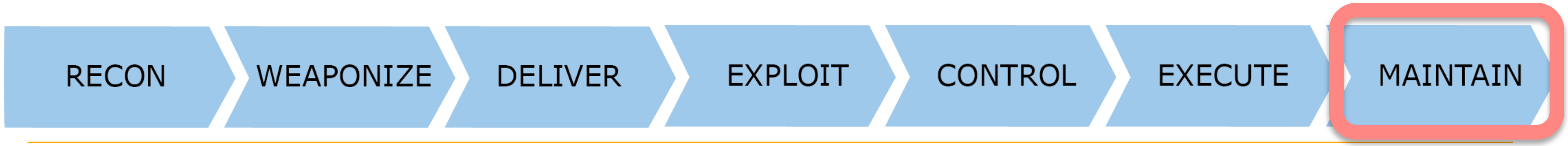
Dissecting Android Malware: Characterization and Evolution

Yajin Zhou
Department of Computer Science
North Carolina State University
yajin_zhou@ncsu.edu

Xuxian Jiang
Department of Computer Science
North Carolina State University
jiang@cs.ncsu.edu

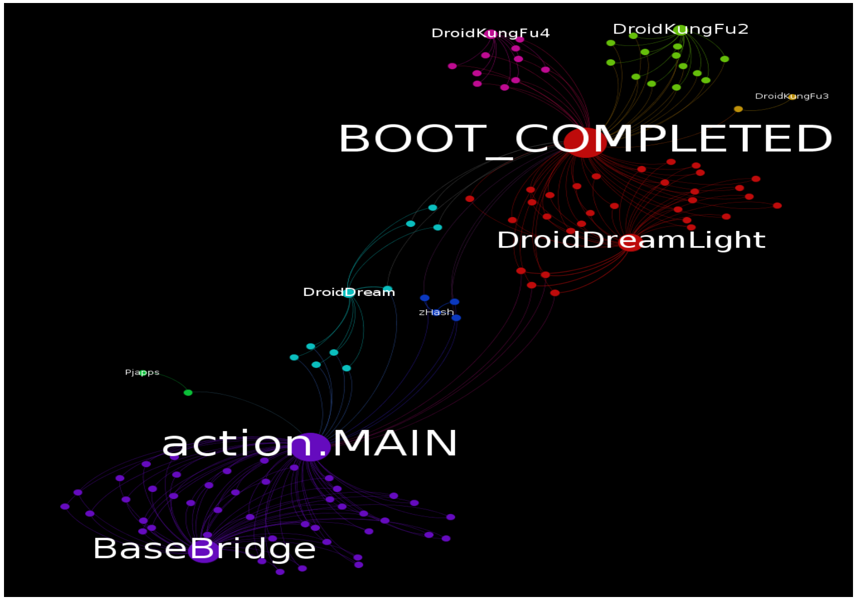
***Abstract*—The popularity and adoption of smartphones has greatly stimulated the spread of mobile malware, especially on**

The goals and contributions of this paper are three-fold. First, we fulfill the need by presenting the first large



Choosing Malware

- Narrowed sample set by filtering on
  - mount –o remount,rw /system
- This was what we thought was indicative of the sample establishing persistence
  - Wrong!!
- Samples were then chosen randomly



Round 1 – Test Tractor Beam Detectors

Sample	Logged		Detected		Detection Result
	Installed	Files Dropped	BOOT_COMPLETED	system.img Change	
1	x	x	x		Success
2	x	x	x		Success
3	x	x	x		Success
4	x		x		Success
5	x	x			Fail
6	x	x	x		Success
7	x	x	x		Success
8	x	x	x		Success
9	x	x	x		Success
10	x		x		Success
11	x	x	x		Success
12	x	x	x		Success
13	x	x		x	Success
14	x	x	x		Success
15	x	x	x		Success
16	x		x		Success
17	x	x	x		Success
18	x	x	x		Success
19	x	x	x		Success
20	x	x	x		Success

Round 2 – BOOT_COMPLETED Detector

		Detected	
		Malware	Non-Malware
Actual	Malware	10	0
	Non-Malware	29	61

Accuracy	71%
Error	29%
Precision	25.6%
Recall	100%

3 .apks were not observed installing – installed to /mnt/secure/asec