A Security Hygienic Smart Charger for Mobile Devices

David Weinstein (The MITRE Corp.)
MoST 2012
Outline

- Motivation
- Challenges
- Prototype
  - Workflow
- Future work
- Questions
What is good hygiene?
...in the computer world?
“An ounce of prevention is worth a pound of cure.”

- Ben Franklin
Personal

Flickr user Odensuke
Vulnerable?
Mobile Security Challenges

- Power constraints
- No practical hardware root of trust
- Lack of virtualization
- Vendors don’t make life easier
- Not a whole lot of downtime
Can we add (security) value to recharging?
Peripheral interface

- USB port provides power and data interface
  - For better or worse
  - Some users plug into work laptop
- Phones are typically idle and unused during this period
- Average person charges once every 2 days
Adding value

- Reboots into custom recovery image
- Measures and checks integrity of hosted device
- Possibly re-flash phone if something is wrong
Measurement and Integrity

- Identify and hash important system components
  - On disk and in memory
  - Kernel and kernel modules
  - System libraries
- Whitelist, not blacklist
Security sold separately?

- Smaller attack surface
  - Special purpose HW/SW based on inexpensive SoC designs
- Lower complexity than commodity OS
- Not constantly network accessible or in attacker’s critical path
Sample workflow

1. User plugs in phone, charger identifies device
2. Charger flashes recovery image and forces boot into it
3. Mounts local storage/filesystems inside recovery image using its own filesystem library
4. Checksum/Signature scan/Rootkit detect
   1. If bad checksum, decrypt/verify signature of gold image
   2. Flash “gold” image
5. Deposit crypto token used to gain temporary access to private network (e.g., VPN)
6. Reboot into fresh image
New measurements

signed/enc module M

plug

adb reboot recovery

mount local NAND

module M

gather hashes

return values

deposit token

reboot

reboot + recovery img self-check

join network, supply token

decrypt/verify module (using server pub key)

compare to known vals
Future work

- Runtime (post-boot) measurement
  - Software timing-based attestation
- Present TPM-like interface?
- Replace use of ADB with something more secure
Questions?
dweinstein@mitre.org