Using Labeling to Prevent Cross-Service Attacks Against Smart Phones

Collin Mulliner, Giovanni Vigna
University of California, Santa Barbara
David Dagon, Wenke Lee
Georgia Institute of Technology, Atlanta

Smart Phones

- Combination of PDAs and mobile phones
- Integrate multiple wireless networking technologies
  - Wireless LAN, Bluetooth, GSM/CDMA/UMTS, IrDA
- Support installation of 3rd-party software
  - For example: VoIP clients, FTP servers, games

Contributions

- Devised Cross-Service Attacks, a new class of attacks against smart phones
- Created a proof-of-concept cross-service attack
- Developed a protection mechanism to prevent cross-service attacks

Introduction to Cross-Service Attacks

- Smart phones integrate different network services
  - GSM, Wireless LAN, Bluetooth, IrDA
- Integration is often done without taking into account the specific characteristics of the different services
  - For example: free vs. pay-per-use services
- An attacker can leverage the interaction between different types of network services
  - For example: gain access to pay-per-use services by exploiting free services

Service Protection

- Local and personal area wireless networking services
  - Devices do not offer comprehensive protection mechanisms
  - Many smart phone applications are developed without security in mind
- Mobile phone services
  - Service providers protect their customers
    - For example: firewalling

Crossing Service Boundaries

- Attack device using local area wireless networking service
  - Exploit insecure configuration of local area wireless networks and networked applications
  - Take control of the device
- Access mobile phone service (**cross service boundaries**)
  - Initiate phone calls or send text messages
  - Exploit pay-per-use services to defraud user
    - For example: 900/0190 calls and/or premium rate text messages

Attack Scenario

- Coffee shop with free wireless Internet access
  - Attacker looks for smart phones joining the wireless network
  - Exploits vulnerable device and causes financial damage

A Proof-of-Concept Attack

- Targets PocketPC-based smart phones
  - PocketPC is the WindowsCE version for smart phones
- Performs buffer overflow/stack-smashing attack against an FTP server
  - Shellcode accesses mobile phone interface and initiates call
- Overcomes complications due to WindowsCE architecture
  - Need to load special DLL for accessing the phone interface
  - Need to guess correct return address

Cross-Service Exploit

Preventing Cross-Service Attacks

- Stack protection (for preventing stack-smashing attacks)
  - Not available or rarely used on mobile devices
  - Does not prevent exploitation of application-logic errors
  - Does not protect against Trojan horses
- Other protection mechanisms needed
  - Detect and prevent attempts to cross service boundaries

Preventing Cross-Service Attacks Through Labeling

- Developed a security mechanism that tracks and controls network interface access using labeling
  - A label indicates contact with a specific network interface
  - A user-defined policy defines which labels should prevent access to a specific network interface
- Labels are assigned to processes as they access network interfaces
- Labels are transferred between processes and files on access or execution

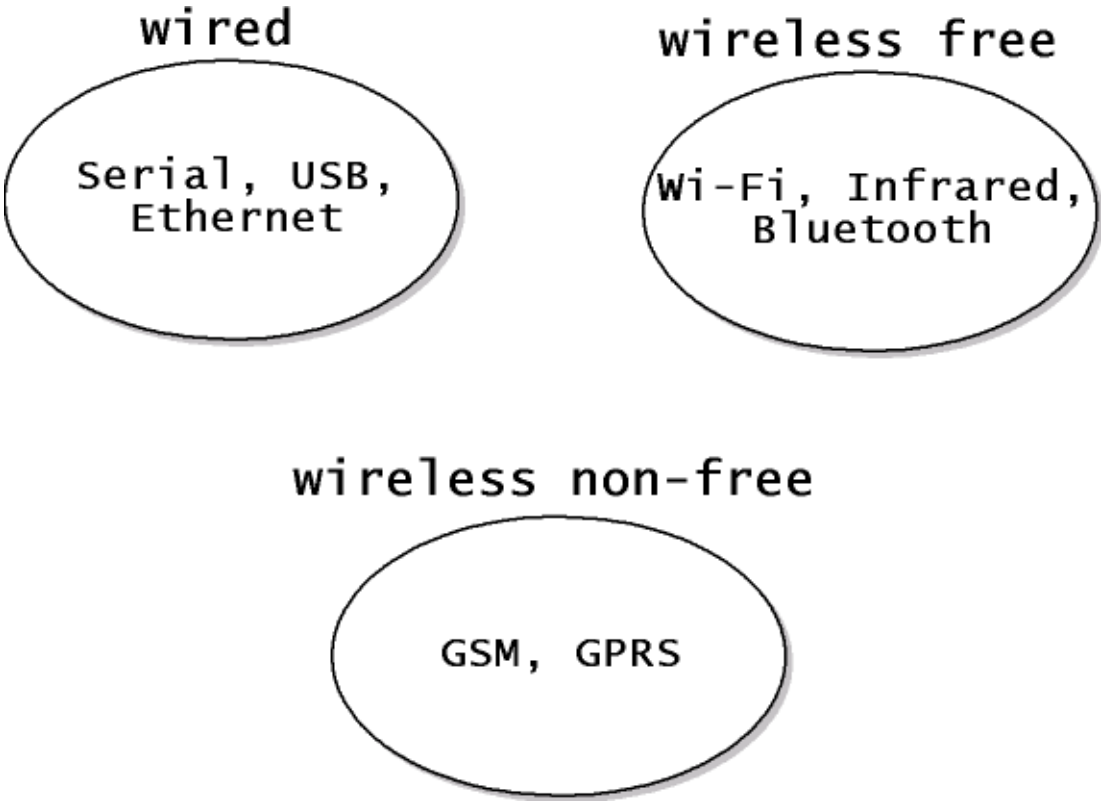
Tracking and Controlling Network Access

- Developed a kernel-level reference monitor
  - Intercepts security-critical system calls
  - Assigns labels to processes and transfers them between processes and resources
  - Enforces access control policies
- Intercepted security-critical system calls:
  - socket(AF_INET, ...) IP-based network access
  - open(...) File and device access
  - execve(...) Program execution

Labeling Processes and Files

- Interface access
  - The process' labels are compared with the access control policy
    - Access is permitted or denied
  - The process is labeled with label of accessed interface
- Resource/file write access and process creation
  - Files and processes inherit labels of creating process
- Resource/file read access and application execution
  - Process inherits labels from accessed and executed file

Label Groups



Access Control and Exception Policy

- Access control rules
  - access <interface> <deny/ask> <label(s)>
  - Example: ***access wireless_nonfree deny wireless_free***
- Exception rules
  - exception <path> <notlabel/notinherit/notpass>
  - Example: ***exception /Windows/activesync.exe notinherit***

Preventing the Attack

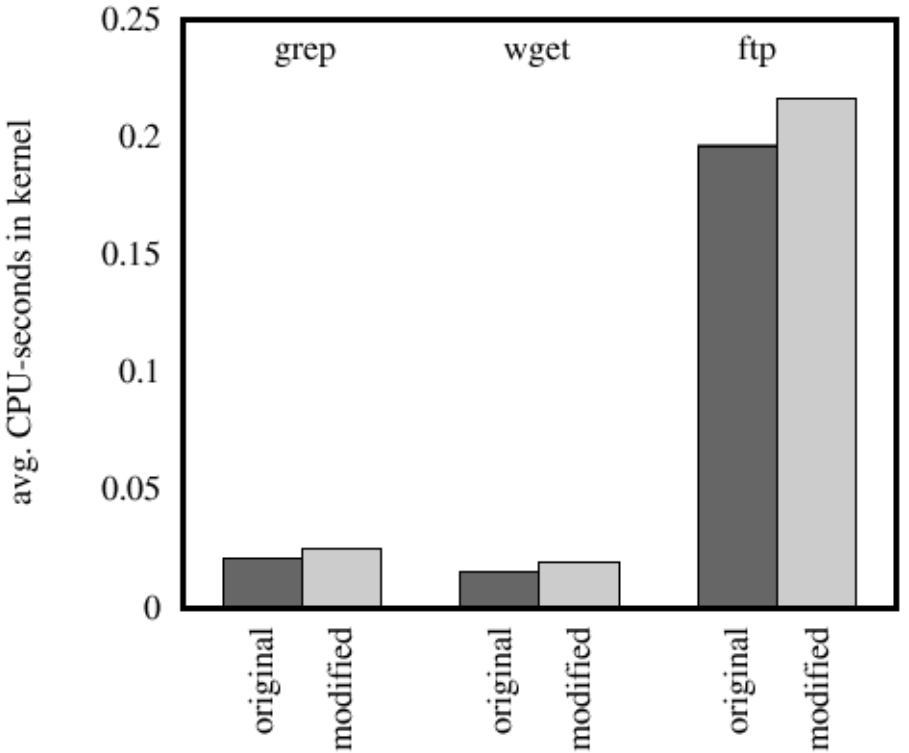
- The FTP server process is labeled on calling socket(...)
  - Label is set for: *wireless_free*
- The exploit tries to access the phone interface
  - For example: open("/dev/ttyS0", ...)
- The reference monitor is invoked
  - Process labels are compared with policy rules
  - The monitor denies access, open(...) returns EACCESS

Evaluation

- Our labeling system effectively prevents attacks that cross service boundaries
- System and policy language are light-weight
  - Appropriate for mobile devices
- Exception rules have to be used carefully
  - Otherwise the labeling system can be bypassed

Overhead

- Reference implementation for Familiar Linux
  - Overhead between 10% and 26%



Conclusions

- Smart phones present new challenges for security designers and analysts
  - Especially the integration of multiple networking services are problematic
- We introduced a new type of attack
- We demonstrated the possible impact of a cross-service vulnerability
- We designed and implemented a solution based on resource labeling

Future Work

- Extend the policy language to support more complex labeling policies
- Improve the implementation of the reference monitor to further reduce overhead

Questions?

Thank you for your attention!