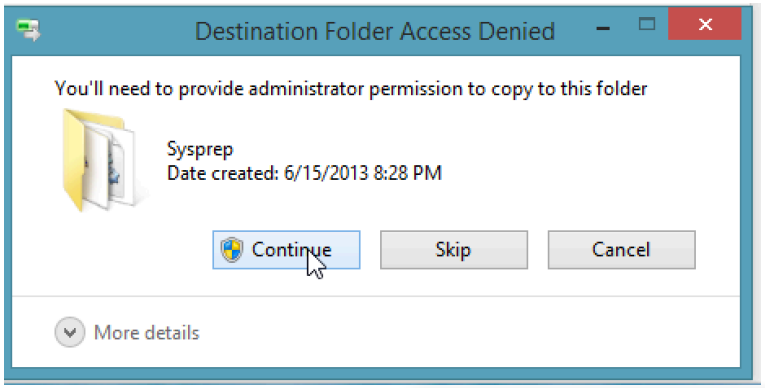
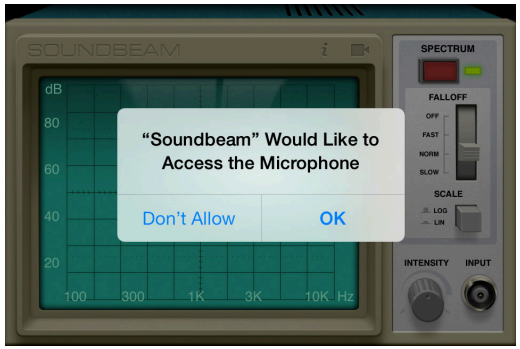
A11y Attacks: Exploiting Accessibility in Operating Systems

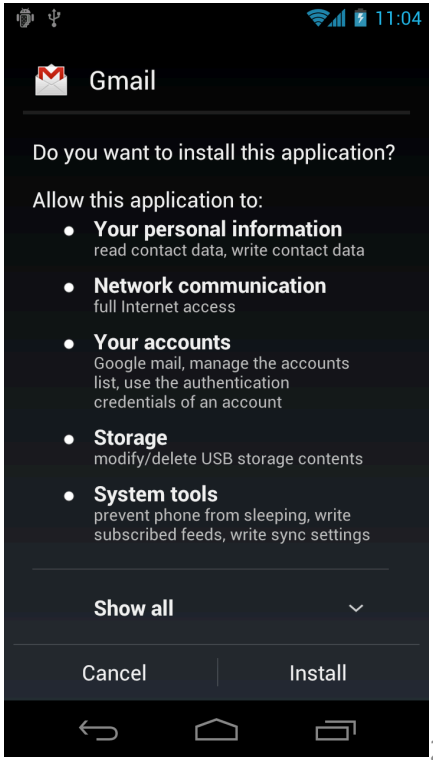
Yeongjin Jang, Chengyu Song,
Simon P. Chung, Tielei Wang,
and Wenke Lee
Georgia Institute of Technology

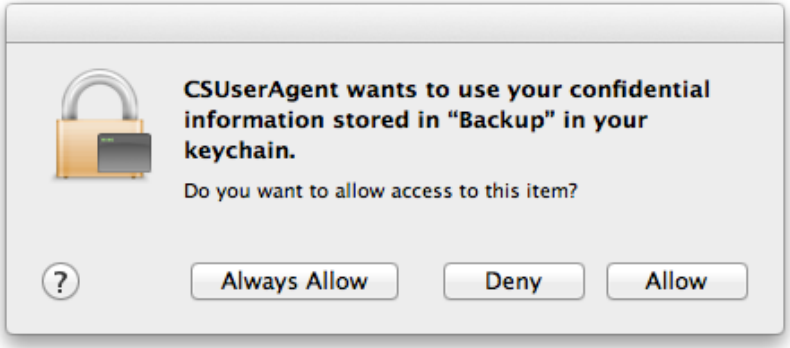
Motivation

- Are these things implemented in secure way?
  - Assumption: input comes from the user





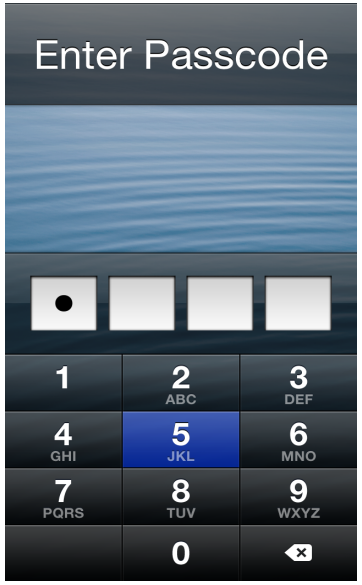


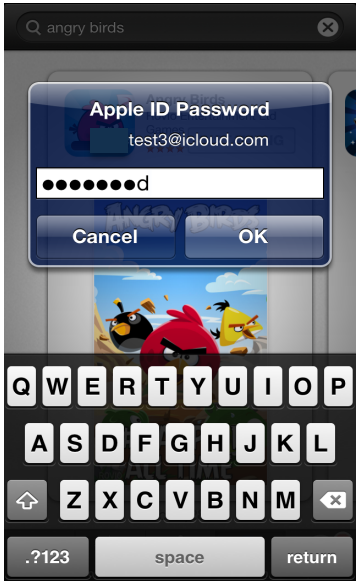


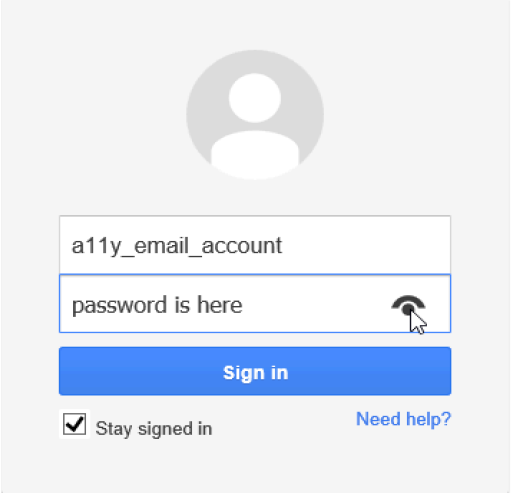
Georgia Institute of Technology

Motivation

- Are these things implemented in secure way?
  - Assumption: output is only visible to the user







Computer Accessibility (a11y)

- For the person with disabilities
  - Visually impaired
    - Text-to-Speech reader
  - Hearing impaired
    - Captioning service
  - Motor impaired
    - Voice Commander
  - Keyboard impaired
    - On-screen keyboard

Accessibility for Everyone

- For ease of access
  - Book reader
    - Text-to-Speech reader
  - At noisy sports bar
    - Captioning service
  - While driving
    - Voice Commander
  - On touchscreen devices
    - On-screen keyboard

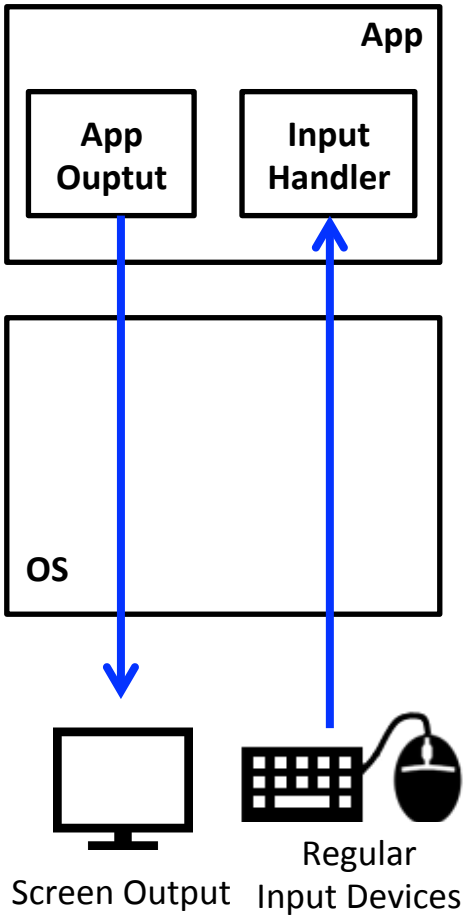
Accessibility Library

- OS opens API for developing A11y features
  - Available in Windows, OS X, Ubuntu, iOS, Android, etc.
- Capability of A11y Library
  - Read UI states of the system
  - Perform actions on UI elements
    - Click
    - settext()
    - etc.

Security Implications of A11y

- Creates new I/O Path
- Break assumptions on I/O
  - Input comes from the user
    - Through a11y interface, a program can send input event to the application.
  - Output can only be seen by the user
    - A11y interface allows to a program can read output of the other applications

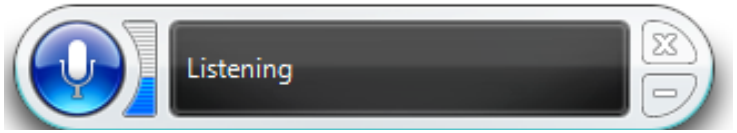
Traditional I/O Paths in OS



A11y Added New I/O Paths to OS

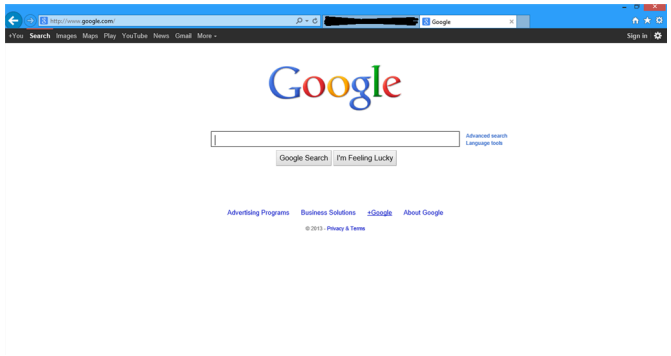
1. User Speaks

2. Voice commander translate it into machine command



1) Click address bar
2) Type google.com

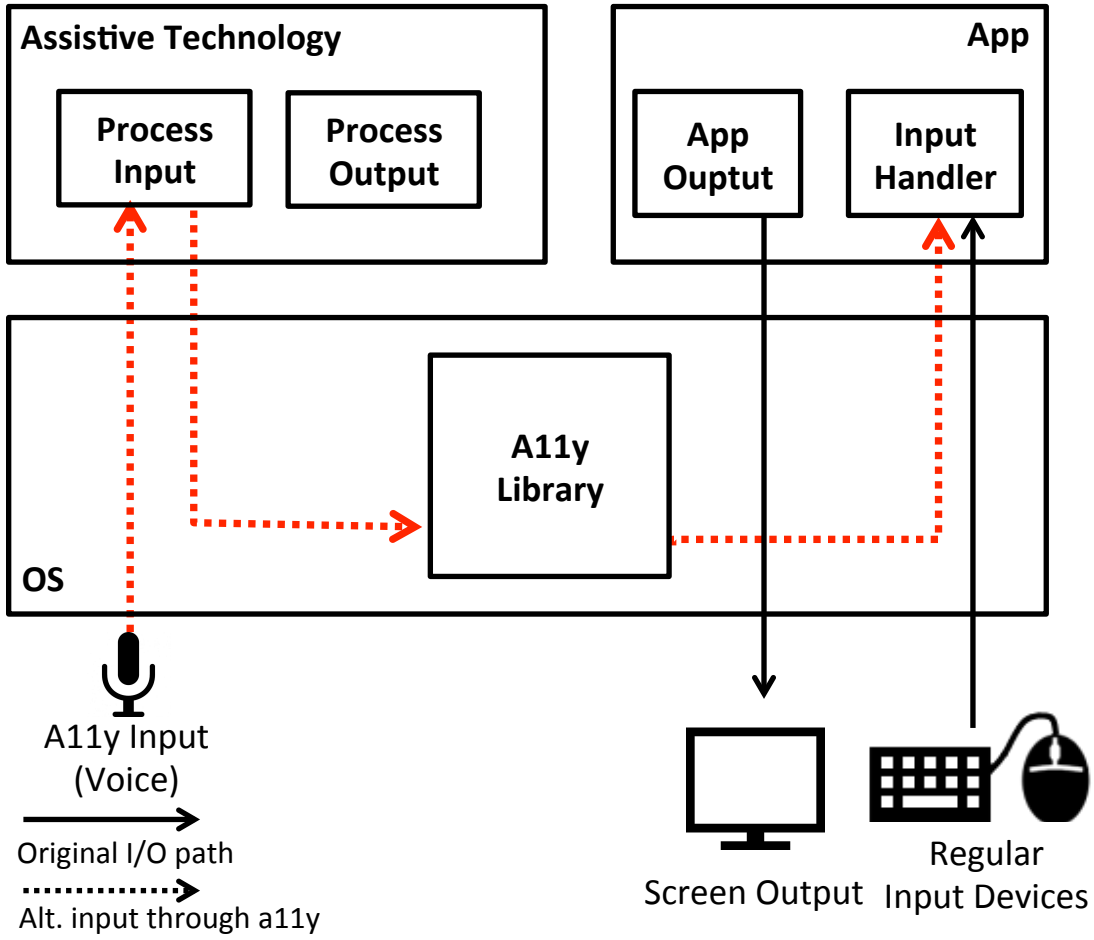
3. OS delivers command to the app (a11y library)



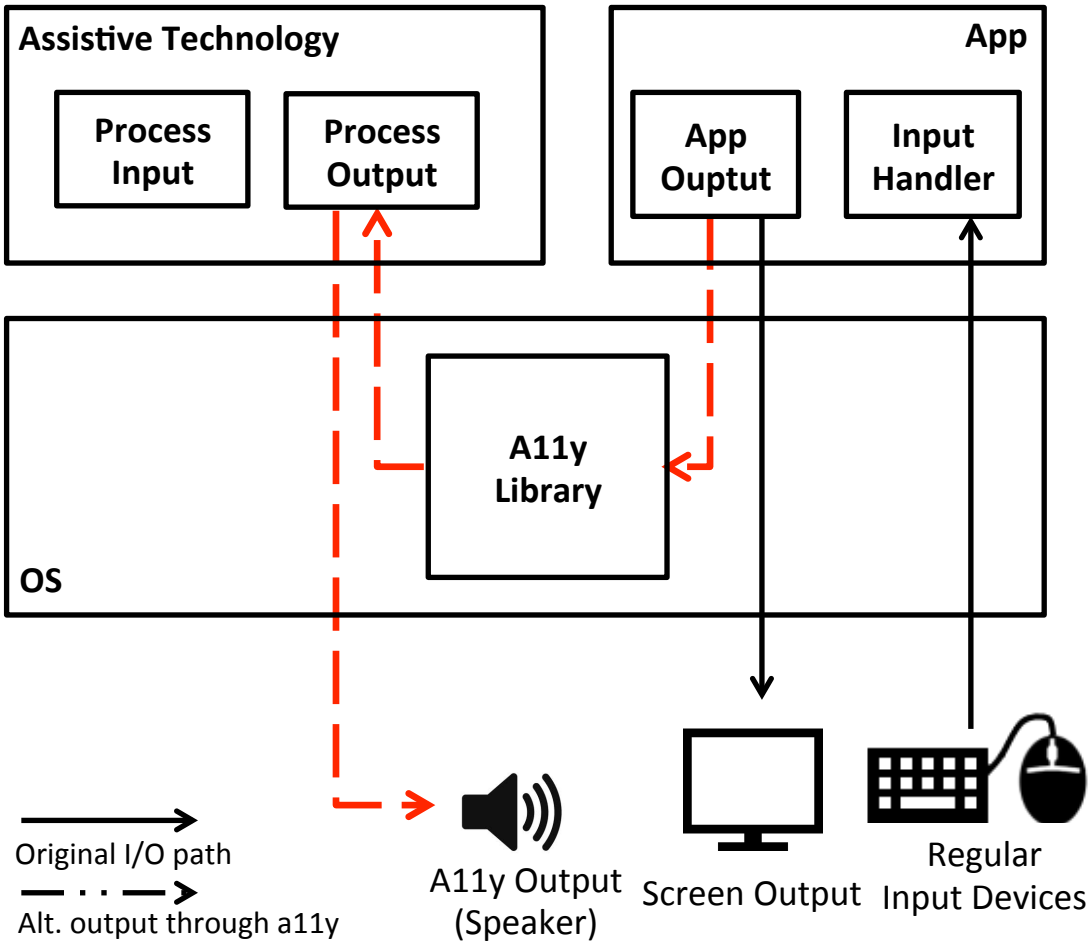
4. App is controlled by Voice

Georgia Institute of Technology

A11y Added New I/O Paths to OS



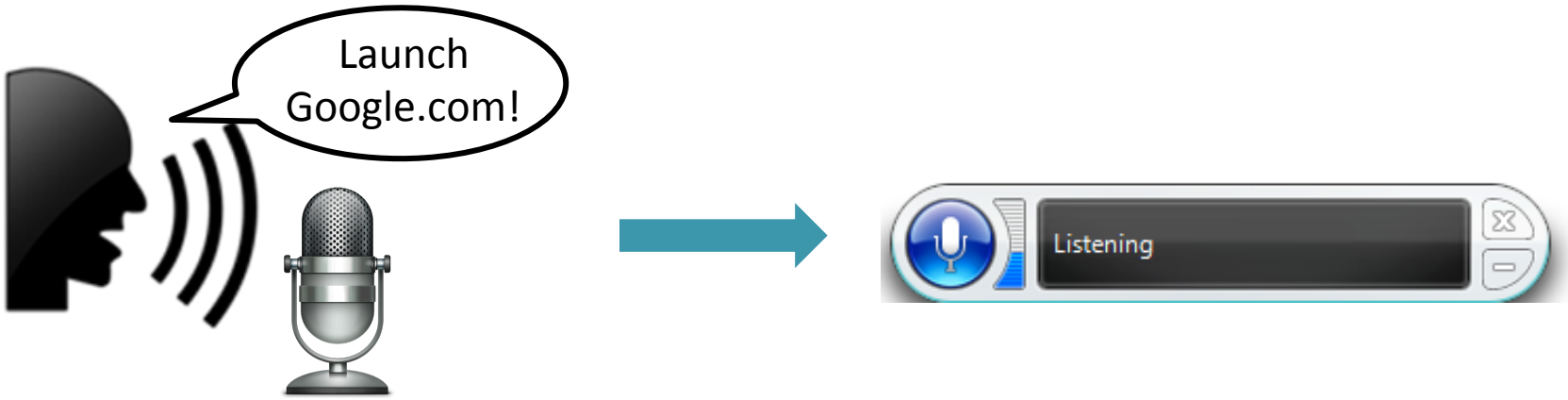
A11y Added New I/O Paths to OS



Required Security Checks

- Security checks must be placed to make the assumption hold
  - Does a11y input really comes from the user?
- Checks can be placed in three different level
  - Assistive Technology (processor of alternative I/O)
  - Operating System
  - Application (protect themselves from alternative I/O)

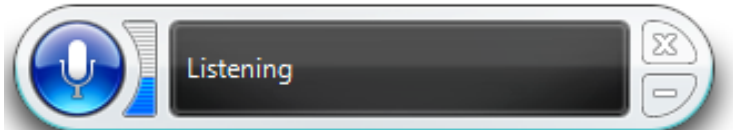
At Assistive Technology (AT) Level



Required checks at AT level

- Is the voice from real human?
  - If not, machine can access it!
- Is the voice matched with registered user?
  - If not, any other human user can access it!

At OS Level

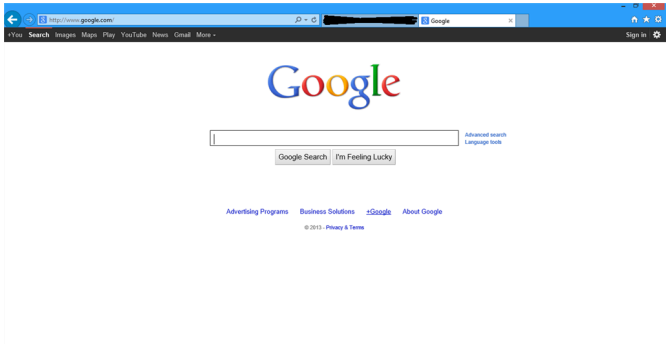


Required checks at OS level

Is this assistive technology allowed to access a11y?

If not, any program (possibly malware) can access it!

At Application Level



Required checks at application level

Should I react to input from a11y features?

Do not allow to perform security sensitive UI actions!

Evaluating A11y Security in OSes

- Objective
  - Check OSs if they are secure under attacks through new I/O path created by supporting A11y
- Method
  - Analyze OS for accessibility features
    - Programmatic access to I/O event
    - Voice commander, password viewer, etc.
  - Test existence of required security checks
    - If not, try to launch an attack

Evaluating A11y Security in OSes

- Target
  - 4 Major OSes
    - MS Windows 8.1, Ubuntu 14.04 Linux
    - iOS 6, and Android 4.4
- Focus
  - Try to evaluate OS default settings
    - AT-level check
      - Voice Commander
    - OS-level check
      - Programmatically controllable I/O
    - App-level check
      - We do not perform the evaluation…

Evaluation on A11y Input

Platform	AT-level check (voice commander)	OS-level Security Check	Vulnerable?
Windows	None (Speech Recognition)	UIPI	YES
Ubuntu	N/A	None	YES
iOS 6	None (Siri)	None	YES
Android	Voice Authentication (Moto X)	User Settings Required	YES

Evaluation on A11y Output

Platform	Reading of UI Structure	A11y leaks on screenshot	Password protection	Vulnerable?
Windows	UIPI	Yes	Yes	YES
Ubuntu	None	No	Yes, but incomplete	YES
iOS 6	N/A	Yes	N/A	YES
Android	User Settings Required	No	User Settings Required	YES

Attacks for missed checkpoint

- We tries to launch attacks if any of security check is missed.
- We found 12 possibly vulnerable points.
  - Windows (3)
    - 2 Privilege escalation, 1 password leak
  - Linux (2)
    - Bypassing process boundary, password leak
  - iOS (4)
    - Bypassing sandbox and authentication
    - Privilege escalation, Password leak
  - Android (3)
    - Bypassing sandbox and authentication
    - password leak

Attacks on Voice Commander

- Voice commander accepts non-human voice
  - Any app capable to play audio can send command
    - Broken assumption: input comes from the user
  - No authentication
    - Windows Speech Recognition
    - Siri
    - Google Now
  - Voice authentication in presence
    - Moto X
      - Vulnerable to replay attack

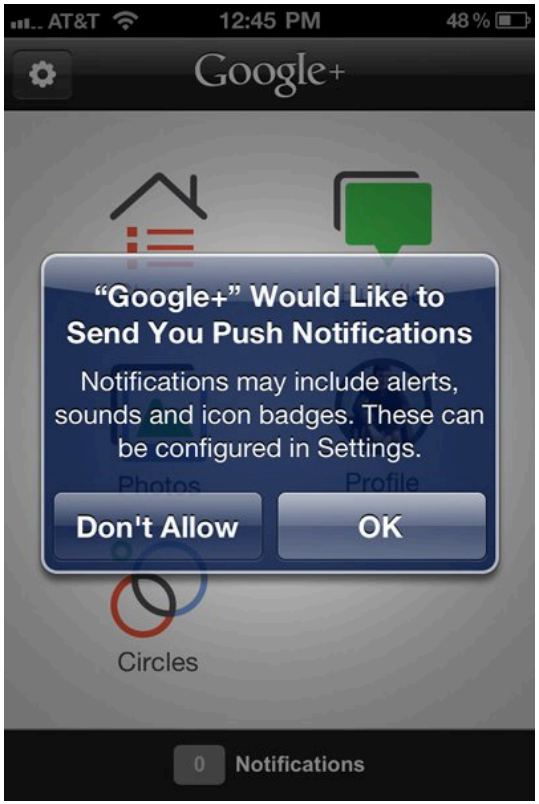
Privilege Escalation in Windows

- Malware runs as normal user can execute Speech Recognition
- Speech Recognition automatically launches with administrative privilege
  - Let A11y user control admin stuffs…
- Malware can get admin privilege by sending voice command to Speech Recognition

Take Control Over Other Apps

- A11y library allows a program send input to the other apps
  - Broken assumption: input comes from the user
- Bypassing app sandbox
  - iOS and Android
    - Sending programmatic input to the target app

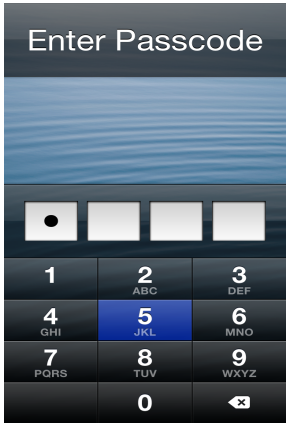
Remote View

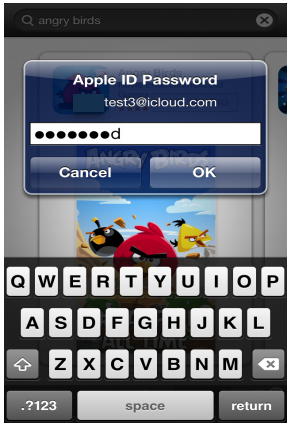

AT&T
12:45 PM
48 %
Google+
"Google+" Would Like to Send You Push Notifications
Notifications may include alerts, sounds and icon badges. These can be configured in Settings.
Don't Allow
OK
Photos
Profile
Circles
0
Notifications


GEMS
X
Confirm Your In App Purchase
Do you want to buy one 24 Gems for $4.99?
Cancel
Buy
BUY!
BUY!

Stealing Password!

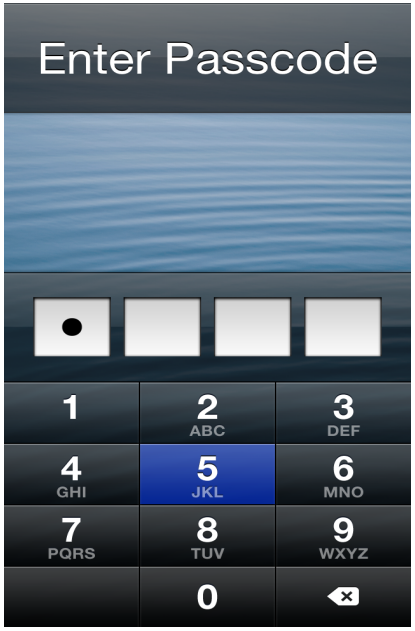
- Visual Feedback
  - Accessibility
    - There is no tactile feedback on touch-screen devices.
    - To reduce *typo*, OS vendors applied visual feedback.
    - Assumes only user can see it.
  - Problem
    - Existing feature breaks its security
      - Screenshot!
        - » iOS6: Private API allows screenshot
        - » Windows: no restriction at all

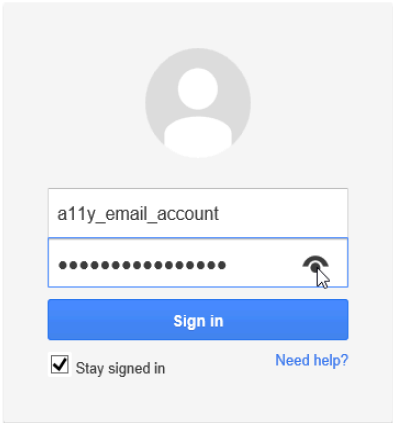




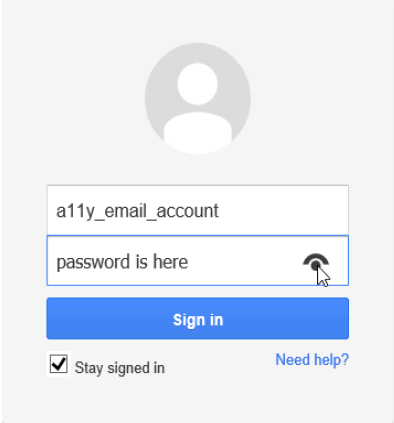
Stealing Password!

- Applying image processing on screenshot leaks password string.



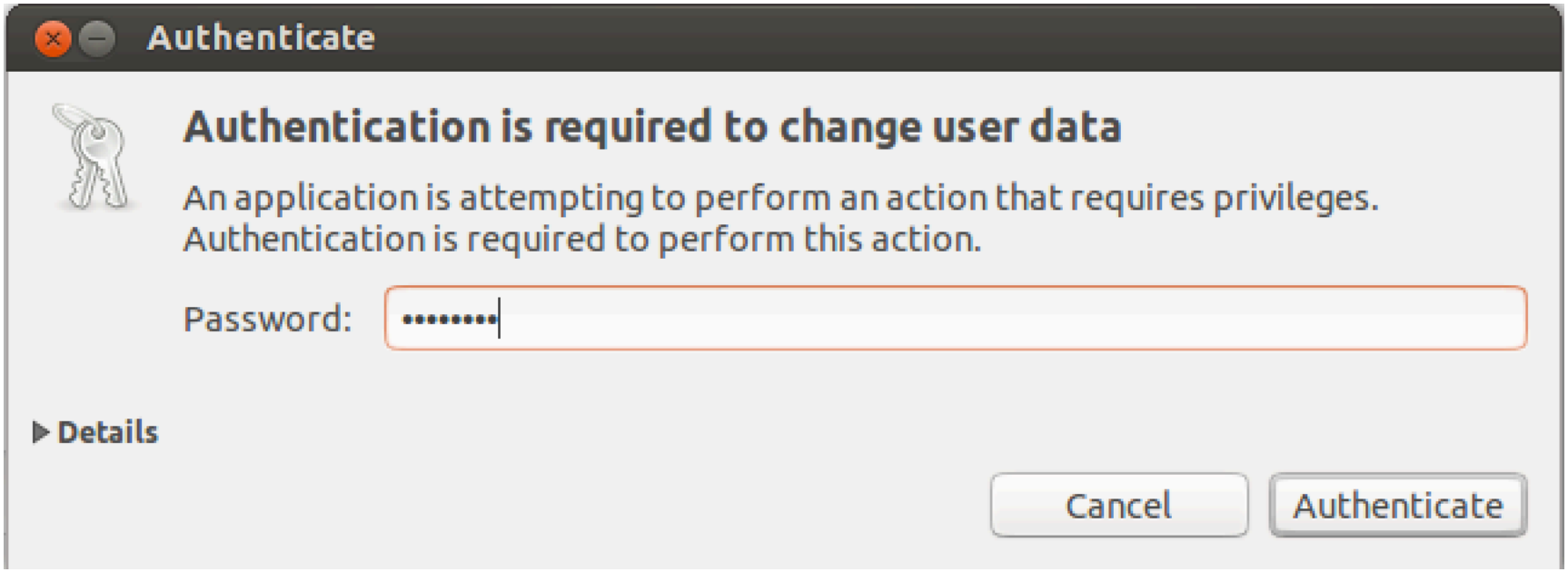


A) Before clicking Eye



B) After clicking Eye

gksudo password dialog



Stealing Password 2

- Two implementation for the same action
  - GTK
    - Pressing Ctrl-C (copy) has no effect on password editbox
      - Security check protects the content
  - ATK
    - Calling gettext()
      - Throws Not Implemented exception
    - Calling copytext()
      - Copy password into clipboard
        - » Missing security checks…

Vendor Responses

- Apple
  - Made UIAutomation inaccessible from regular apps
    - Requires special permission to access the library
  - Disabled private API for taking screenshot
- GNOME ATK
  - Acknowledges finding as vulnerability
  - Tries to fix in GNOME 3.14

Vendor Responses

- Android
  - AccessibilityService is their feature…
    - "*Does not consider these feature requests as vulnerabilities*"
- Windows
  - Does not consider UAC bypass as security vulnerability

Discussions

- Root-cause
  - Maximizing Compatibility
    - The UI is expected to run as if it gets the real input on a11y request
    - Programmatic input processed as same as the real one

	Real Touch Click	**A11y Click**
Intermediate func	onTouchEvent()	performA11yActionInternal()
Final handler in UI	performClick()	performClick()

Discussions

- Root-cause
  - Problems when it handled differently
    - On gksudo dialog, copytext() works while Ctrl-C does not work!
    - New implementation could miss security checks.

```
GTK::CopyText() {
        if(text->isVisible)
                return text
        else
                return null;
}
```

```
ATK::CopyText() {
        …
        return text
        …
}
```

Discussions

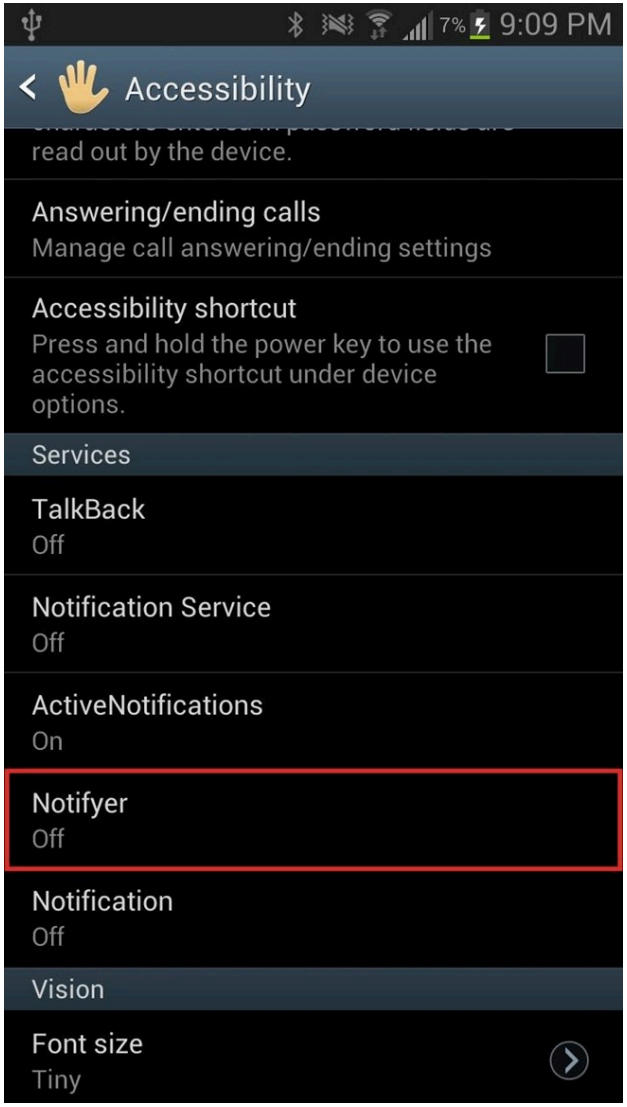
- Root-cause
  - No correct authentication for alternative input
    - Any program can send fake voice…
  - Technical & economical difficulty
    - Possible solution for voice authentication
      - Liveness check
      - Challenge-response
    - Practical issues
      - Processing power
      - Power consumption
      - etc

Discussions

- Root-cause
  - Weak access control on a11y libraries
    - Windows: None
    - OS X : None
    - Ubuntu: None
    - iOS 6 : None -> patched in iOS 7
    - Android: User settings
      - Not enough…

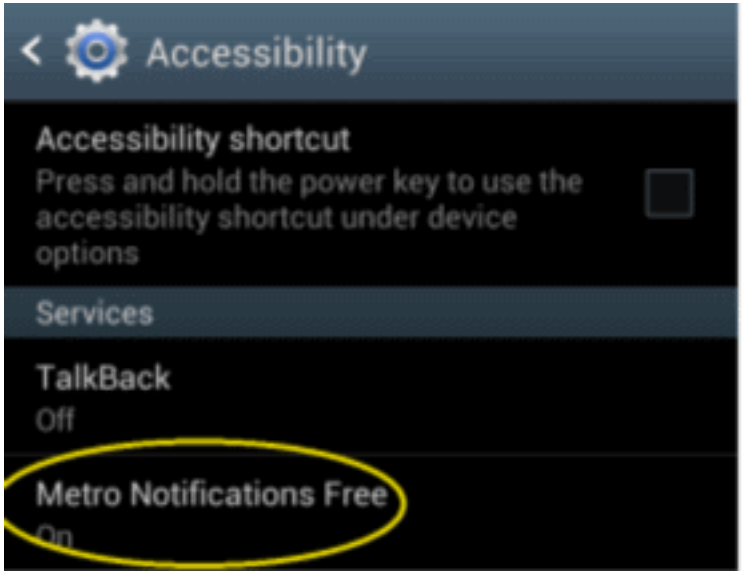
Android

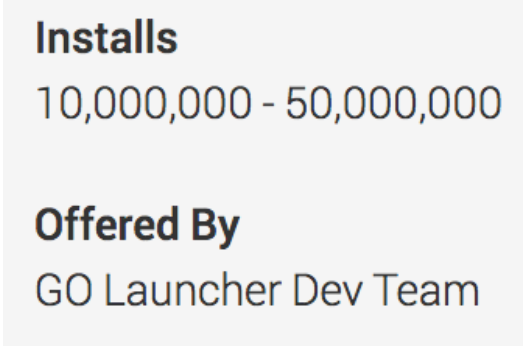
- Settings for a11y
  - AccessibilityService is available upon user config
    - User must set the application as accessibility service



Android

- App uses A11y
  - A non-assistive technology uses a11y
    - For supporting restricted UI features…
    - Downloaded more than 10 million times…





Georgia Institute of Technology

Discussions

- Recommendations
  - Apply access control on a11y library
  - Provide mechanism to distinguish a11y I/O from the real I/O requests
  - For the security sensitive UIs, get input from physical devices, and not others.

Conclusion

- Accessibility in OS
  - Supporting accessibility creates new I/O path to the OS
  - Security mechanisms in OS has trust in I/O
    - The assumption would not valid if a11y I/O treated as same as the real one
  - We found 12 attacks…
  - OSes need to be design a11y securely against these attacks.

Questions?

- Q&A