CSE 551 Group 2: Mobile Phone Information Leaking
For our project we looked into information leaking on three Android/Windows/iPhone and created an app for the Windows 7 phone to test its vulnerabilities.

The specific vulnerability we were trying to test was that of user information leaking by the phone.

We currently have the first release of this app which shows our current progress.
How does an Android application get private data?

- Apps by default do not have permissions
- Every app when it is installed will tell you the permissions it needs
- Most important for information leaking:
  - **Network Communication** – *full internet access*
  - **Your personal information** - *read contact data*
    - These two permissions allow Apps to send your data to remote servers
How does an iPhone application get private data?

- Apps must ask user for permission to access GPS data, but not for contact data
- Users do not have access to permissions lists like on Android
- Must choose apps that are installed wisely
SMobile reports that ~20% of applications on Android are capable of accessing private data
- Potential for abuse is present
- Similar numbers for iPhone were found
- Less data for Windows Phone 7 market

Study done by Intel found that many popular (trusted) applications send user location data to advertisers
Securing Private Data

- Problems:
  - Most apps use private data to enhance the user experience properly
    - Permission restriction just makes it harder, but cannot stop private data abuse
  - Hard to monitor HOW an application uses private data
    - TaintDroid created by Intel tries to accomplish this
  - Finding the right balance between security and access is better dealt with on a user by user basis
Securing Private Data

- What is currently in place?
  - Marketplaces for Android, iPhone, and Windows applications all must be approved
  - Developer information must be provided
  - Android
    - Apps must have permissions (done at installation) to access private data
  - iPhone
    - Users must give permission during runtime for GPS data, but contact data is less protected
  - Windows 7
    - Does not allow users to have direct access to data. Must go through a black boxed function to retrieve data. User must approve of the access during runtime.
Securing Private Data

- Third party solutions available
  - Lookout
    - Provides security suites for protection against spyware/malware
  - TaintDroid
    - Monitors HOW the data is used rather than dealing with permissions on accessing data
- Security issues will always be present if apps are able to grab user data
  - Selective on which apps are being installed
  - Check for unneeded permissions
    - Don’t always just click “allow”
Isolated Storage

- Feature allows applications to create app-specific local storage
- Has been in use since Silverlight version 2
- Isolated storage data is completely separate from OS storage and storage from other apps
- All I/O operations are restricted to isolated storage
Isolated Storage

- Any information shared between applications cannot be local to either app
- Shared information is stored online and apps then can create local copies if needed
- Read and writes can be done using the IsolatedStorageFile (text, images, other) or IsolatedStorageSettings (name/value pairs) classes
Isolated Storage
There is no limit to the size of isolated storage files so developers must use space responsibly.

This security measure attempts to protect valuable data by restricting application access to any data except for its own isolated storage files.
Launchers

- Since applications run in an isolated state they do not have access to phone and OS features.

- Launchers are API's that allow applications to perform the various tasks that are not readily available.
Launchers

- **EmailComposerTask**: launches the Email application and displays a new email message.
- **MarketplaceDetailTask**: launches the Windows Phone Marketplace client application which then shows the details page for a product specified by the unique identifier you provide.
- **MarketplaceHubTask**: launches the Windows Phone Marketplace client application.
- **MarketplaceReviewTask**: launches the Windows Phone Marketplace client application which then displays the review page for your application.
- **MarketplaceSearchTask**: launches the Windows Phone Marketplace client application which then shows the search results based on search terms you provide.
- **MediaPlayerLauncher**: launches the [Media Player](#) application and plays the media file you specify.
- **PhoneCallTask**: launches the [Phone application](#) and displays the specified phone number and display name.
- **SearchTask**: launches the Search application and performs search query you provide.
- **SmsComposerTask**: launches the Messaging application which displays a new SMS message.
- **WebBrowserTask**: launches the [Web browser](#) and displays the URL you provide.
When a launcher is called the application is tombstoned then reactivated upon the completion of the launchers task

Tombstoning is a process where applications are terminated when a user starts another process. The state information is saved so that it can resume where the user left it. This allows multitasking style functionality without the performance compromise.
Choosers

- APIs very similar to launchers, main difference is that choosers return values back to the calling application

- To use launchers or choosers developers must reference the phone tasks in their code using Microsoft.Phone.Tasks
Choosers
Every application that is posted to the Marketplace must pass a security inspection and adhere to a list of requirements before being made available to the public.
Marketplace Testing

- General Procedure
  - Create, test, debug app
  - Obtain publishing license
  - Upload app to AppHub for validation/certification
  - If it passes the app will be added to the Marketplace
Certification Requirements

- App must be reliable (in accordance with published best practices)
- Apps must execute efficiently and not negatively impact performance
- Apps must not interfere with phones functionality (cannot change settings without first notifying the user)
Certification Requirements (contd.)

- App must be free from viruses and malicious software
- Code must handle ALL exceptions and must compile under Retail configurations
- App must start and become responsive within 20 seconds
Marketplace Testing

- Certification Requirements (contd.)
  - App must not use more than 90 MB of data at any time
  - App must not pass any user identifiable or personal information to another program, service, or application without user permission and without giving an explanation on how the information will be used
Marketplace Testing

- Certification Requirements (contd.)
  - App can only use the Windows Application Platform documented APIs
  - App must be packaged as a XAP file upon submission
Windows Phone 7

- Mobile operating platform developed by Windows.
- Successor to the Windows Mobile platform aimed at the consumer market.
- User interface attempts to integrate the operating system with 3rd party apps and Microsoft’s other services.
Security goals

- Apps built in Silverlight on top of OS.
- No interaction between app code and underlying software.
- Apps each have their own separate storage.
- Phone communication handled by apps that ran on top of your apps.
In early January some Windows Phone 7 users began to receive messages from their service provider that they were getting close to their monthly data limit.

On further inspection the phones were sending data even when not in use.

Microsoft and bloggers began to look for the source of this ‘phantom data’.
Eventually Rafael Rivera (a blogger) discovered that Yahoo Mail! was the source of the extra data. This was confirmed by Windows a few hours later.

A temporary solution was proposed to decrease how often the Yahoo client searched for updates and new messages.
Recreating scenario

- There were some difficulties with recreating the previous scenario.
- Some of the more popular apps get a special developers kit with more access to the OS.
- This problem was with how imap packages were handled by the Yahoo Mail! server. We do not have access to e-mail capabilities.
For our app we have tried to remove as much user information from the phone.

One way is to do this covertly so that the user doesn’t know it’s happening.

Second way it to trick the user into willingly giving over information.
Our goals for this application were:

- See how the Windows Phone 7 stores and protects user information.
- See how much of this information an app can copy.
- Test out other ways in which we could get this user information.
For the app what we’re basically doing is collecting information and then storing it into the isolated storage for that app.

We then use WP7 Isolated Storage Explorer to retrieve the file into which the information has been written.
Application : Demo

page title
So far these are our conclusions:
- The WP7 environment is made safe through restrictions on developers.
- The marketplace filters out most malware.
- Most of the threat will come from user.
- This will be in the form of sideloading apps.
- The other threat is users giving away their information.