Building Secure Mobile Apps

Sergey Gorbaty
@ser_gor

Martin Vigo
@martin_vigo

Martin Vigo

Product Security Engineer

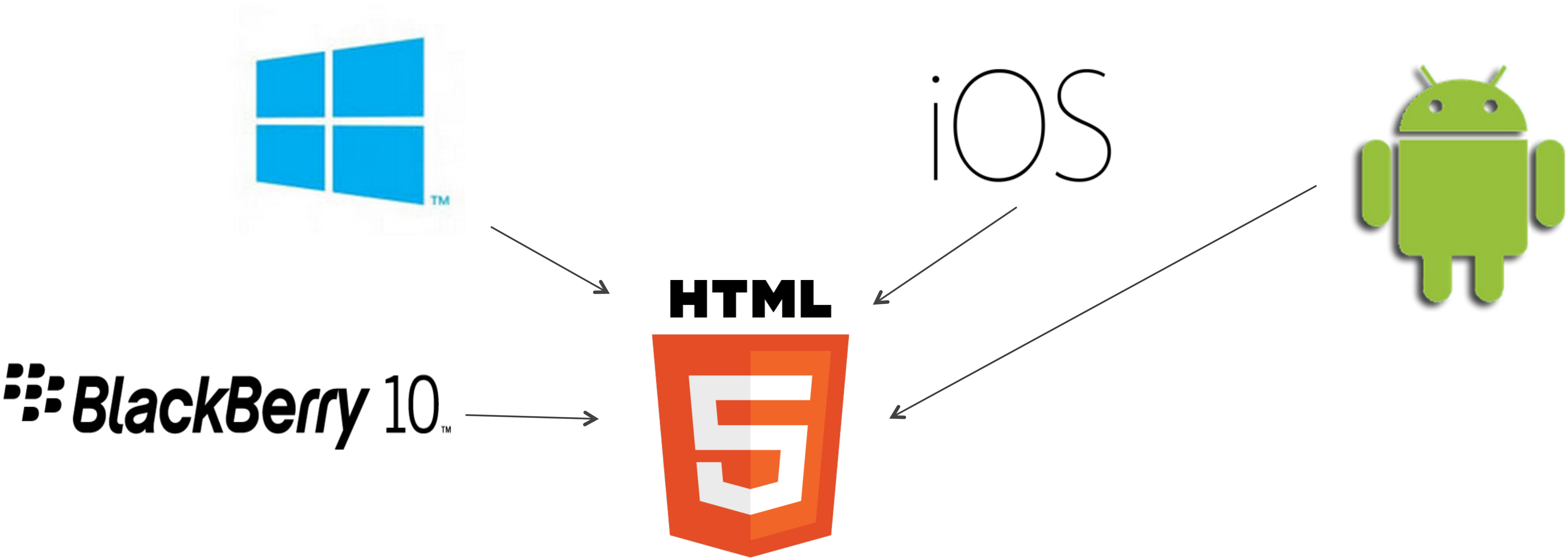
Sergey Gorbaty

Senior Product Security Engineer

Outline

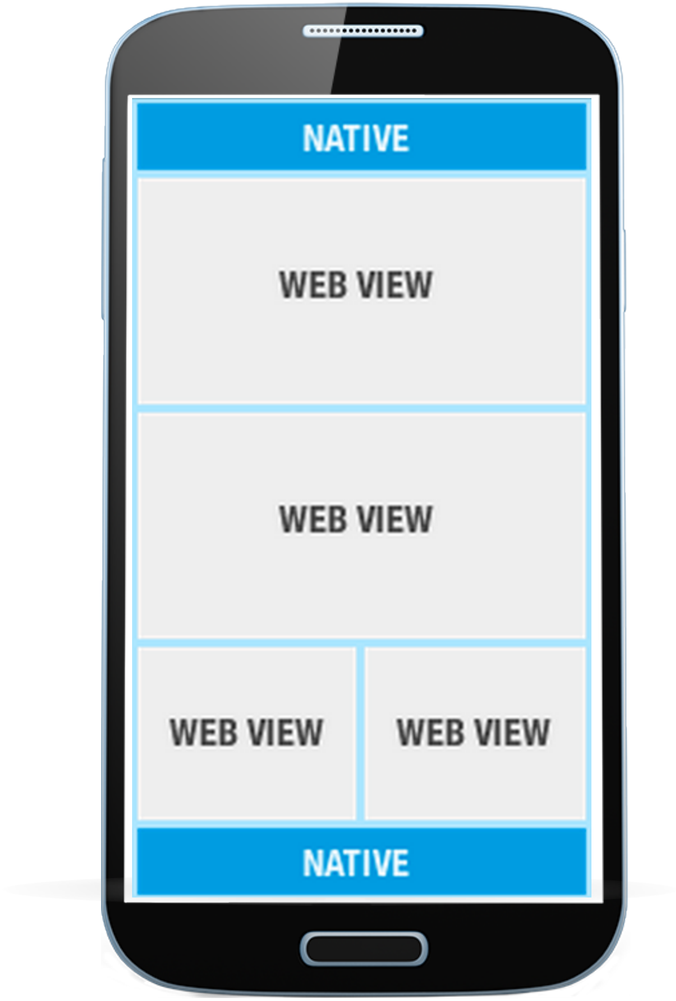
- Attacks on Mobile Apps
- Developing Secure Mobile Apps
- What Frameworks Help You With
- Demos

Attacks on Mobile Apps



Mobile App Threats

- Native Mobile App Threats
  - File system, DB Storage, Logs
  - Network Communication
  - Clipboard
  - Backups
  - RPC, URL scheme handlers
- Web App Threats
  - Input validation
    - Session management
    - Web app logic flaws
  - Web vulnerabilities
    - XSS, CSRF
    - Injections
      - SQL, header

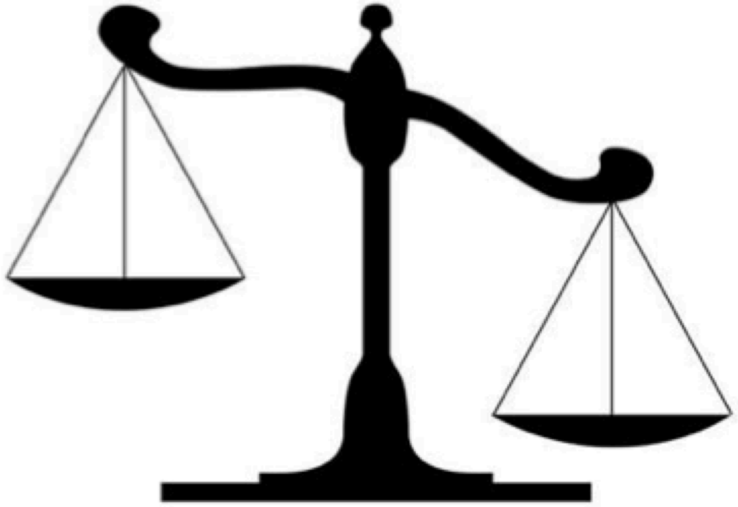


Outline

- Attacks on Mobile Apps
- Developing Secure Mobile Apps
- What Frameworks Help You With
- Demos

Developing Secure Mobile Apps

- iOS/OS X ‘Secure Coding Guide’
  - Comprehensive, 120 pages long
  - Covers topics from buffer overflows to web vulnerabilities
  - https://developer.apple.com/library/iOs/documentation/Security/Conceptual/SecureCodingGuide/SecureCodingGuide.pdf
- Android.com ‘Security Tips’
  - 6 articles on
    - Storing data
    - Using permissions
    - Using networking
    - Using RPC
    - Webview security
  - http://developer.android.com/training/articles/security-tips.html



File System

Excessive Logging

- Explicit logging
  - Debugging
  - Feedback
  - Analytics
- Automatic logging
  - Generic information
  - Exceptions

Excessive Logging - TODO

- Do not log credentials including username, password, and OAuth tokens
- Do not log emails, names, titles, company information
- Do not log hardware ids including IMEI, UDID
- Prefer to log internal opaque IDs if possible
- Disable logging before shipping

Hardcoded Secrets

- Encryption keys
- PINs
- Settings
- Credentials

Hardcoded Secrets - TODO

- Don’t hardcode ANY secrets
- Query secrets only when necessary
  - Don’t keep them in memory longer than needed.
- Do not assign secrets to global variables
- Disable autocorrect on sensitive fields

Insecure storage

- Explicit storage
  - Data
  - Preferences
  - Logs
  - Crash Reports
- Automatic storage
  - Temp Files
  - Cache

Insecure storage - TODO

- Use secure storage for secrets
  - Keychain
  - AccountManager
- Verify that no sensitive data is stored without your knowledge
- Control App flow and encrypt data when device is in background or locked

Automatic Caching

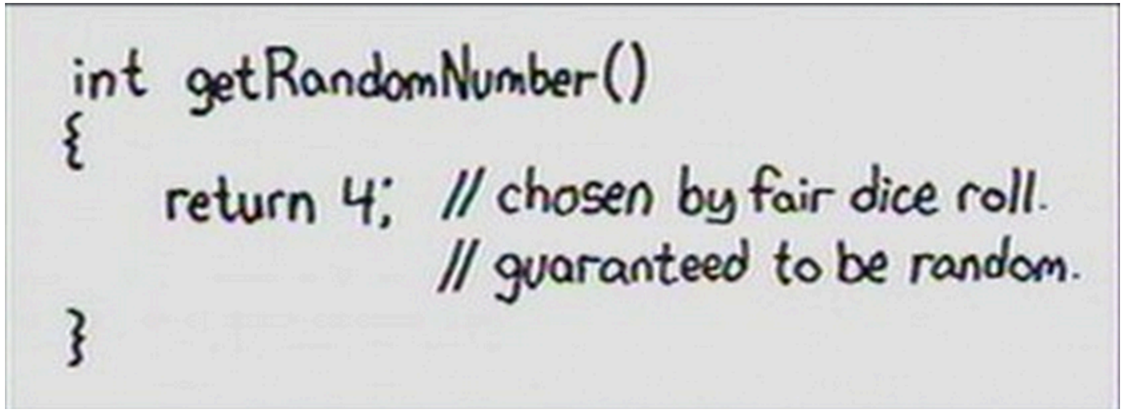
- Databases
- Preference files
- Plists
- Logs
- Requests and responses

Automatic Caching - TODO

- Double check what is being cached
  - File system explorers
  - Database managers
- Prevent network requests caching
  - ‘Cache-control: no-cache, no-store’
  - Disable web view disk caching
  - Use in-memory caching only
- Destroy Cache data on logout

Encryption

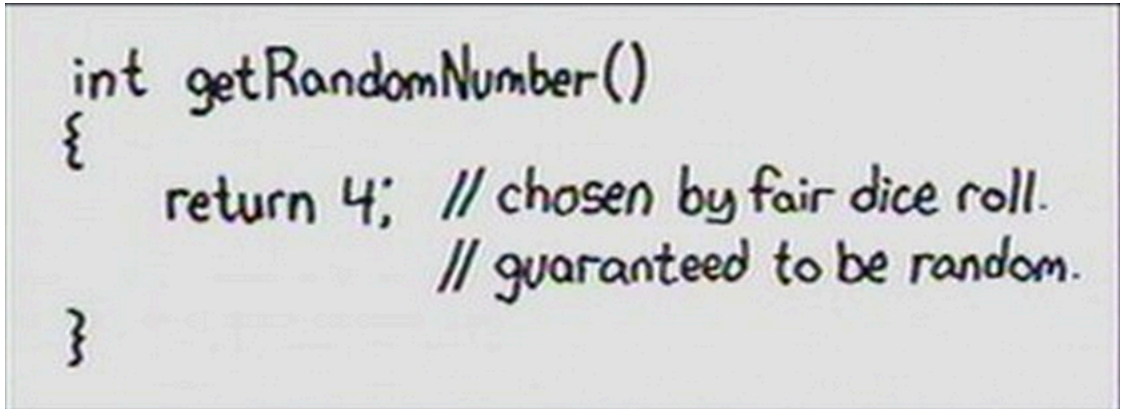
- Do we need encryption?
- Types of Crypto
- Personal implementation
- Performance



```
int getRandomNumber()
{
    return 4;  // chosen by fair dice roll.
               // guaranteed to be random.
}
```

Encryption - TODO

- Encrypt customer data stored on the device and removable media
  - Use AES 128 bit or stronger
  - Never use ECB mode
- Use Key Derivation for encryption key
  - PBKDF2 (10000 rounds, SHA 256 or higher)
  - bcrypt
  - scrypt
- Passcode Protection
  - Store it hashed
  - Use SHA-256 + secure random generated salt
  - Store salted hashes of passcode in secure storage
- Use PIN for additional entropy

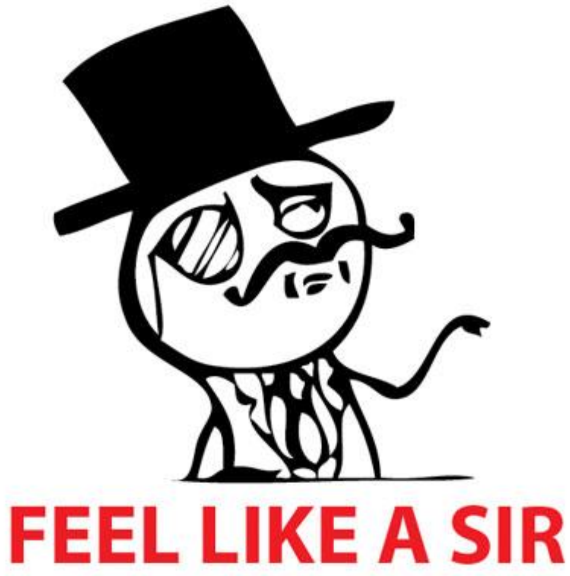


```
int getRandomNumber()
{
    return 4;  // chosen by fair dice roll.
               // guaranteed to be random.
}
```

Network Communication

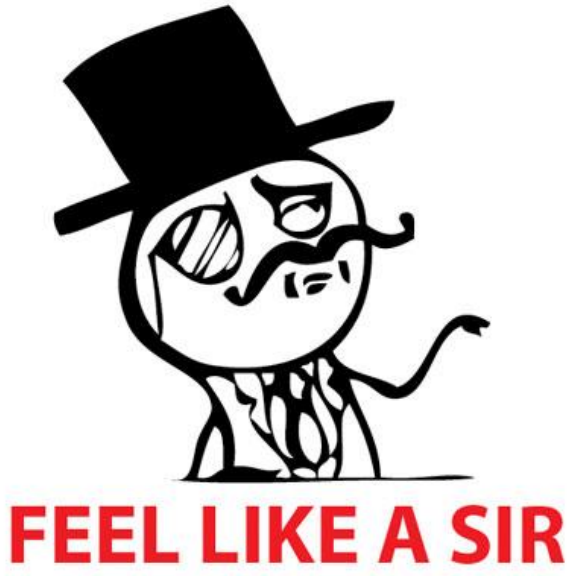
Protocols

- Use of encryption layer?
- All endpoints covered/secure?
- Cyphers supported
- Default cyphers
- Caching



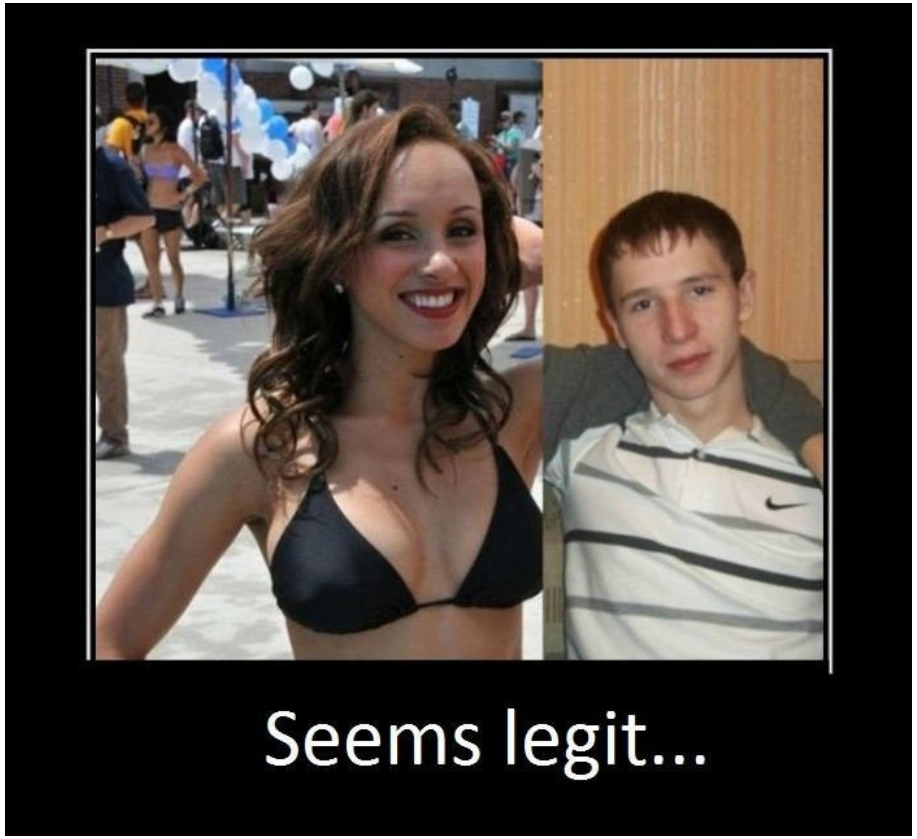
Protocols- TODO

- Do not implement SSL/TLS trust validation bypasses
- Use SSL3/TLS1.x
- Disable caching containing sensitive data



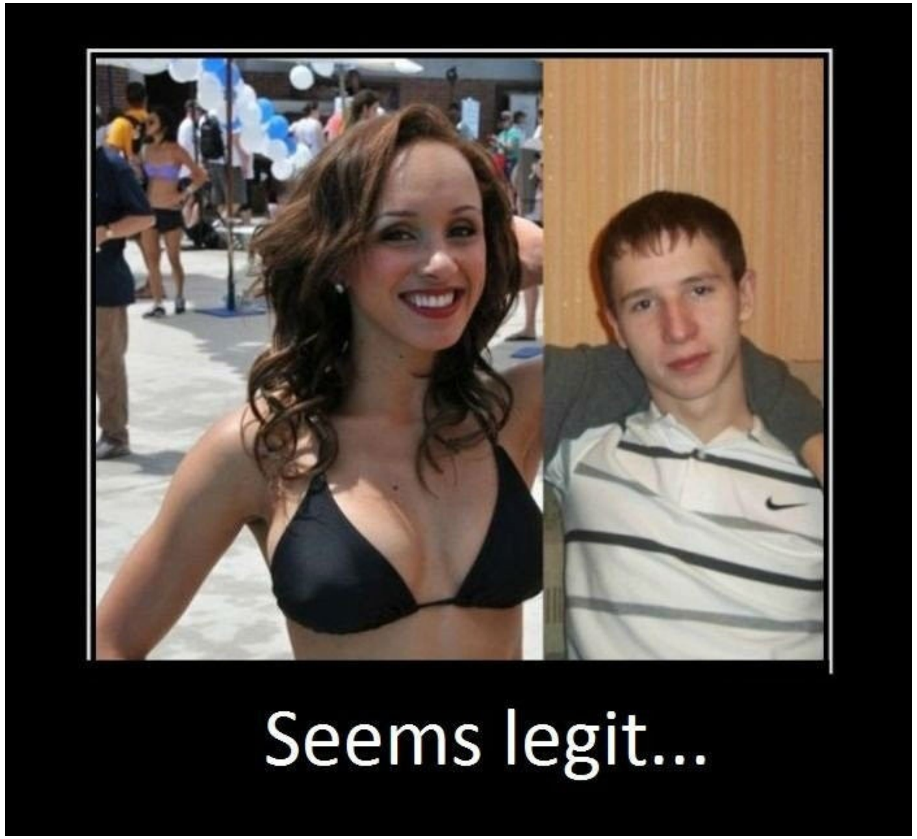
Certificates

- Self-signed
- Invalid
- Certificate validations
- Bypass



Certificates - TODO

- Don’t allow self-signed certificates
- Validate all certificates
- Never bypass Certificate Authority root of trust



Session management

- Logout
- Expiration
- Data destruction

Session management - TODO

- Implement inactivity timeouts to prompt user to re-login after prolonged inactivity
- Implement business logic for logout
  - Delete all associated data
  - Expire the session on client AND server side
- Protect your Cookies

Clipboard

Clipboard

- What data can make it to the clipboard?
- Who can access the it?
- Is there any security layer?

Clipboard - TODO

- Clipboard is not a secure method of information exchange
- Clipboard can be accessed by any application
  - At any point in time
  - Without user prompt
- Limit the data available to Clipboard
  - Don’t allow sensitive data

Backups

Backups

- What data is backed up
- Encryption
- Access limitations

Backups - TODO

- Filter what data can be backed up
  - NSURLIsExcludedFromBackupKey
  - android:allowBackup
- Backups are not a secure storage
- Create backups and explore them for sensitive data

Screenshots

Screenshots

- What can be captured
- Automatic screenshots
- Any way to set limitations?

Screenshots- TODO

- Prevent users from taking screenshots of sensitive data
  - *getWindow().setFlags(LayoutParams.FLAG_SECURE, LayoutParams.FLAG_SECURE);*
- Prevent automatic caching in iOS
  - *willEnterBackground* API
  - Use splash screen

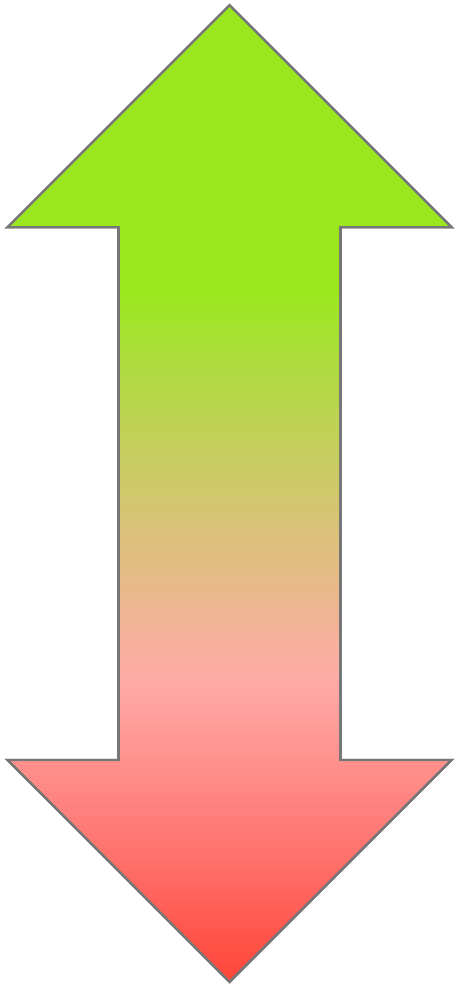
Outline

- Attacks on Mobile Apps
- Developing Secure Mobile Apps
- What Frameworks Help You With
- Demos

Mobile Frameworks

The breakdown

- **All** focus on rapid development using HTML
- **Most** provide easy ways of creating secure TLS connections
- **Fair amount** provide authentication support
- **Few** provide secure credential storage
- **Very few** provide secure data storage



Hybrid Apps

- Can access device internals through plugins
  - Camera, photos
  - Accelerometer, GPS, Compass, Gyroscope
  - Keychain
  - SD card
  - Etc.

Frameworks Security

WebView

- Additional Threats
- JavaScript support
- Framework specific security requirements

WebView - TODO

- Third party scripts shouldn’t be trusted
- Iframe sandboxing
  - Don’t include script in the context of application
- Whitelist specific domains and paths
  - Avoid wildcard (*) whitelist
- Minimize the number of exposed plugins

Outline

- Attacks on Mobile Apps
- Developing Secure Mobile Apps
- What Frameworks Help You With
- Demos

Demo

Looking at files inside Apple Sandbox - iExplorer

Demo

XSS with BEEF on Hybrid mobile app

Protecting Mobile Apps

What to focus on

- Follow best development practices
  - Brush up on OWASP top 10 mobile threats
  - Review official vendor recommendations
  - Follow recommendations for storing secrets and data
  - Exercise minimal logging
  - Using TLS
  - Use security frameworks, don’t roll your own crypto
- Use free security assessment tools
  - HTTP traffic examination: Burp Suite, Fiddler, Charles Proxy
  - App sandbox examination: iExplorer, drozer, Android debugging bridge
  - Source code review: Findbugs, Brakeman, Scanjs

THANK YOU!

Sergey Gorbaty
@ser_gor

Martin Vigo
@martin_vigo