Mobile Application and BYOD (Bring Your Own Device) Security Implications to Your Business

Dmitry Dessiatnikov
DISCLAIMER

All information in this presentation is provided for information purposes only and in no event shall Security Aim be liable for any direct, indirect, incidental, or other special damages however caused arising in any way out of the use of information in this presentation.
Who Am I?

- President at Security Aim
- Specializing in web, mobile and network security assessments
- Penetration tester with web development and database administration background
- Salt Lake OWASP Chapter Leader
- Board Member UtahSec.org
- CISSP
- PCI QSA/ASV
Agenda

• Background – why should we care?
• BYOD – how is your business exposed?
• Tool release
• Demo – compromise Android phone
• OWASP TOP 10 Mobile Risks
• Examples of common attacks
• Demos – compromise iOS application
• Conclusions
Why should we care?

Many Organizations Will Not Provide Devices

- **Today**: 6%
- **2016**: 45%
- **2020**: 15%

Source: Hunting and Harvesting in a Digital World: The 2013 CIO Agenda, Jan 1 2013

* Gartner analyst estimates

n=2053 worldwide

Why should we care?

According to Gartner: “Through 2014, employee-owned devices will be compromised by malware at more than double the rate of corporate-owned devices.”

http://www.gartner.com/technology/topics/byod.jsp
Why is mobile a concern?

- Typically weak passwords because of not user friendly keyboard
- Mobile devices are online longer and taken more places because most users want to be reachable by phone
- Easier lost/stolen than desktops
- Mobile device defenses are immature
- Legitimate market for spyware
Why is mobile a concern?

• Mobile network providers configure devices to prefer wi-fi hot spot over cellular data to get users off their network
• If wi-fi is not turned off device is attempting to connect automatically to saved SSIDs
• Mobile application session tokens do not expire for a long time
Why is mobile a concern?

• While servers/PCs are often protected with firewall/AV/HID mobile devices are not
• Limitations of reviewed mobile AVs:
  – limited by sandbox
  – can't hook to system calls
  – can only do static code analysis and check for signatures of known malware
• Email – spam filtering/virus protection/anti-phishing
• SMS is the new agent for virus/spam/smishing
Tool release – SE-SMSer

- Automates the process of sending out text messages with a trackable social engineering link
- Used for mobile social engineering assessments
- Uses Google Voice™ communications service, the registered trademark of Google Inc.
- Requires Google account credentials and access to the Google Voice™ communications service.
- Available at www.securityaim.com/resources
Tool release – SE-SMSer

$ ruby SE-SMSer.rb
Usage: SE-SMSer.rb [-iuposthvy]

SE-SMSer options:
  -i, --input=FILENAME  File containing one target phone number per line.
  -u, --username=USERNAME  Your Google account username
  -p, --password=PASSWORD  Your Google account password
  -o, --output=FILENAME  File containing the phone numbers of targets to be social engineered and the hashes of those phone numbers that will be used to identify victims.
  -s, --sesite=SESITE  Social Engineering site URL without http://. The link created will be URL/[shortened MD5 hash of the email address]
  -t, --timelag=TIMELAG  Pause in seconds between sending each text
  -h, --help  Show this message.
  -v, --version  Show version.

$ ruby SE-SMSer.rb -i targets.txt -u your_username@gmail.com -p your_password -o victims.txt -s securityaim.com -t 1

"Sending SMS to:
"Waiting for 1 seconds before sending the next SMS"
"The total number of sent text messages: 1"
Tool release – SE-SMSer – Remote compromise of a non-rooted Android phone

DEMO
Why is mobile a concern?

- As consumers we assume that the manufacturers of the mobile devices care about security of their customers’ data and resources
- OS: Google, Apple, Microsoft, Nokia, etc.
- OEM: Apple, Samsung, LG, Microsoft, etc.
- MNO: Verizon, AT&T, T-Mobile, Sprint, etc.
Android Specific Security Concerns

• For Android Open Source Project – the most common operating system in the world:
  – AOSP 4.0+ security features:
    • ASLR (Address Space Layer Randomization),
    • DEP (Data Execution Prevention)
    • On-device Encryption
  – OEM becomes the weaker link and focus of attacks
  – Purchased device has the latest firmware?
Android Specific Security Concerns

• Out of the box Android phones come with pre-loaded applications

• Security of pre-loaded applications:
  – Installed by both OEMs and MNOs
  – Have default permissions not explicitly accepted by the users
  – Reviewed by security professionals?
  – Expose devices and data
iOS Specific Security Concerns

- Apple Picking
- Additional functionality as “Siri” has security implications
- Default settings allow “access when locked” to:
  - Siri
  - Passbook
  - Reply with message
- Siri Proxy
Why is mobile application security a concern?

• Lack of security training for mobile application developers

• Commonly outsourced

• Corporations exposed through unsecured services required for mobile applications to connect back
OWASP Mobile Security Project
Top Ten Mobile Risks

M1: Insecure Data Storage

- Lost/stolen device or malware infected
- Developers assume that users will not have access to the device file system
  - Credentials
  - Cookies
  - Location data
  - UDID/EMEI, Device Name, Network Connection Name
  - Personal Information: DoB, Address, Social, Credit Card Data
- Application Data:
  - Stored application logs
  - Debug information
  - Cached application messages
  - Transaction histories

M1: Insecure Data Storage

Credit: iGoat – Ken van Wyk (ken@krvw.com), Sean Eidenmiller (sean@krvw.com)
KRvW Associates, LLC
M1: Insecure Data Storage
M2: Weak Server Side Controls

- Pressures for fast mobile deployment
- Applies to backend services
- Corporate environments exposed:
  - Insecure APIs and web services
  - Mobile clients are trusted
  - Lessons from web application security forgotten
M3: Insufficient Transport Layer Protection

- Typical mobile application - client to server data exchange
- Data traverses multiple networks often without user/developer knowledge:
  - Carrier network
  - Internet
  - WiFi
- Often SSL/TLS is not implemented properly or used only during authentication
M3: Insufficient Transport Layer Protection

Credit: iGoat – Ken van Wyk (ken@krvw.com), Sean Eidenmiller (sean@krvw.com) KRvW Associates, LLC
M3: Insufficient Transport Layer Protection

DEMO
M4: Client Side Injection

• Mobile application clients are trusted
• SQL Injection
• XSS
• Multi-user applications
• Shared device
• Paid-for-only content
M4: Client Side Injection

Goat Hills Picayune
Fair and Balanced

Search all free-to-read articles...

Free: Area Man Outraged
Free: Weather-Predicting Cat

Search all free-to-read articles...

"or "1w1"

Free: Area Man Outraged
Free: Weather-Predicting Cat
Premium: Mayoral Twitter Sc...
M4: Client Side Injection

DEMO
M5: Poor Authorization and Authentication

• Making security decisions based on device specific identifiers that can’t be revoked:
  – Phone number
  – IMEI
  – IMSI
  – UUID

• Assume hostile mobile platform

• Use of identifiers that can be easily spoofed
M6: Improper Session Handling

• Longer expiration times or non-expiring mobile sessions
• Predictable session tokens/low entropy
• Session fixation
• Inability to expire tokens in case of lost/stolen devices
• Device identifier used as session token
M7: Security Decisions Via Untrusted Inputs

- Bypass security controls/models
- Sensitive actions should require re-authentication
- iOS – URL Scheme allow Safari to make phone calls or send SMS
- Android – Abusing Intents
- iOS Skype app – using XSS to make calls
M8: Side Channel Data Leakage

- Developers love to collect data including sensitive data
- Data footprint is often unmanaged on mobile device:
  - Keystroke logging
    - Cut and paste
    - Autocomplete
    - Backgrounding
  - Crash can be caused to send sensitive data to system logs and sent off for troubleshooting
  - Web caches
  - Screenshots
M8: Side Channel Data Leakage

DEMO
M8: Side Channel Data Leakage

Credit: iGoat – Ken van Wyk (ken@krvw.com), Sean Eidenmiller (sean@krvw.com) KRvW Associates, LLC

© Copyright 2013 Dmitry Dessiatnikov - SECURITY AIM
M9: Broken Cryptography

- Improper implementation of strong crypto libraries
- Home grown crypto implementations, obfuscation, encoding, serialization
- Store key with encrypted data
- Applications use SSL but don't require a valid certificate
- Invalid certificate handling - ActiveSync
M10: Sensitive Information Disclosure

• Mobile application code can be reverse engineered
• Hardcoded passwords in mobile application code
• Private API keys stored on the client
Conclusion

• Be aware of the risks before you make significant time and financial investment
• Secure mobile application development training and testing is critical
• Don’t make assumptions about security
• To know if your mobile platform, framework, application is secure test it!
Q & A

Dmtry Dessiatnikov

dd@securityaim.com

Twitter: @SecurityAim