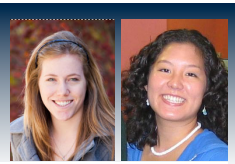
Analyzing WebView Vulnerabilities in Android Apps

Stephanie Rogers, Erika Chin, and Professor David Wagner



COLLEGE OF ENGINEERING, UC BERKELEY

Introduction

We perform a large scale measurement study to determine how many Android applications may be vulnerable to malicious websites that a user may access while browsing.

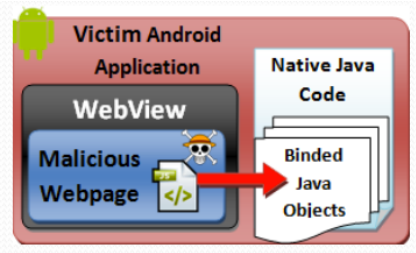
Android Overview

WebView – allows a developer to display web content within their own app

Pros:
- Allows for easier interaction with a website

Cons:
- Could allow JavaScript to invoke application code[1]
- Website has access to system resources and data[1]

WebView-Based Attacks



Threat Model[1]

EXAMPLE

Mobile App code:

```
myWebView.addJavascriptInterface(new
MobileClass(), "Mc");
```

Web app code:

```
<script>
   Mc.mobileFunction(x,y,z)
</script>
```

Approach

Android Application-centric:
- Identify overly premissive features of WebViews

Website-centric:
- Can a malicious website be displayed in the WebView?
- Crawl the page looking for ads

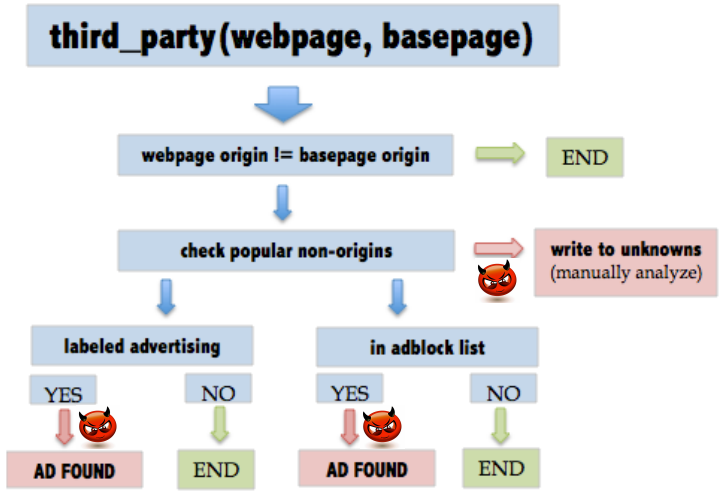
My Contributions

(In Progress)
- Build a tool to determine whether a user can navigate away from a designated webpage and land on a potentially malicious third-party site
- Measure the accuracy of the tool

(Future)
- Manually analyze a portion of the applications
- Create case studies that illustrate the impact and dangers of WebViews that can be controlled by malicious websites

Web Crawler

Tool: build a program that takes as input a URL and then build a crawler to determine if that website includes third-party content (specifically ads)



Challenges

- Identifying ads
- Determining if a site is not of the same origin
- Dealing with websites of non-origin that have never been classified

Conclusion

- Android developers need to be aware of these attacks when using WebViews
- We measure the prevalence of these WebView-based vulnerabilities

[1] Source: "Attacks on WebView in the Android System," T. Luo, et al., ACSAC 2011.