Breaking and Fixing Origin-Based Access Control in Hybrid Web/Mobile Application Frameworks

Martin Georgiev    Suman Jana    Vitaly Shmatikov

The University of Texas at Austin
Once Upon A Time ...
Java
C#
C/C++
Objective-C
Woulda been nice if...
Woulda been nice if...

- Platform-independent
- Portable
- Reuse old Web apps
- Low maintenance
Woulda been nice if...

- Platform-independent
- Portable
- Reuse old Web apps
- Low maintenance

Hybrid App Model
Woulda been nice if...

- Platform-independent
- Portable
- Reuse old Web apps
- Low maintenance

Over 400,000 developers worldwide

Hybrid App Model
Problem:

How to equip Web apps with access to native resources?
The World Of Hybrid Frameworks
The World Of Hybrid Frameworks

WebView

Java
C/C++/C#
Objective-C

bridge

appcelerator
appMobi
APPSPRESSO

bridge

APACHE CORDOVA
MoSync
Sencha

...
The World Of Hybrid Frameworks

WebView

Same Origin Policy

Java
C/C++/C#
Objective-C

appMobi
appcelerator
APPSPRESSO
Apache Cordova
MoSync
Sencha

bridge
bridge
bridge
...
The World Of Hybrid Frameworks

Same Origin Policy

WebView

HTML

JS

bridge

Java
C/C++/C#
Objective-C

OS Access Control Policy

appMobi
appcelerator
APPSPRESSO
COCOON
Apache CORDOVA
MoSync
Sencha

...
The Hybrid Security Model
Attack Model

WebView

myapp.com

HTML

JavaScript

CONFIDENTIAL
What Is The Origin Of This JavaScript Object?
Attack Model

WebView

myapp.com

HTML

Answer: It Does Not Have An Origin!
Attack Model

WebView

myapp.com
HTML

<iFrame src="AttackersR.us">

Answer: It Does Not Have An Origin!
Answer: It Does Not Have An Origin!
Answer: It Does Not Have An Origin!
Answer: It Does Not Have An Origin!
Answer: It Does Not Have An Origin!
Attack Model

WebView

myapp.com

<iFrame src="AttackersR.us">

Answer: It Does Not Have An Origin!

All Your Secrets Are Belong To Us!

Developer provided whitelist

CONFIDENTIAL

[Diagram of a web view with security mechanisms and character icons]
Attack Model

WebView

myapp.com

<iframe src="AttackersR.us">

All Your Secrets Are Belong To Us!

TO THE CITY

CONFIDENTIAL
Bridges to Native Resources

addJavascriptInterface  
prompt / alert / confirm

loadUrl / executeScript / InvokeScript  
cookies

Home-made
Bridges to Native Resources

- `addJavascriptInterface`
- `prompt` / `alert` / `confirm`
- `loadUrl` / `executeScript` / `InvokeScript`
- `cookies`

Home-made

VULNERABLE
Bridges to Native Resources

- wrong regular expressions
- wrong URL interception
- lack of origin propagation
- frame confusion
- wrong scheme
Prevalence

WRITE_CONTACTS

WRITE_EXTERNAL_STORAGE

READ_CONTACTS

ACCESS_FINE_LOCATION
100% API compatibility
0.24 - 5.8% overhead
No code changes in existing apps
Same mechanism for all bridges / all platforms
They Are Fixing It Now!

"Adding a SecureToken [NoFrak] is a good idea and we should implement this"

dev@cordova.apache.org
It's 11pm. Do you know if your hybrid framework is secure?

- No way for the app developer to restrict access to native resources (unlike PhoneGap)
- No internal framework defenses, either
- All apps built on these frameworks are generically insecure