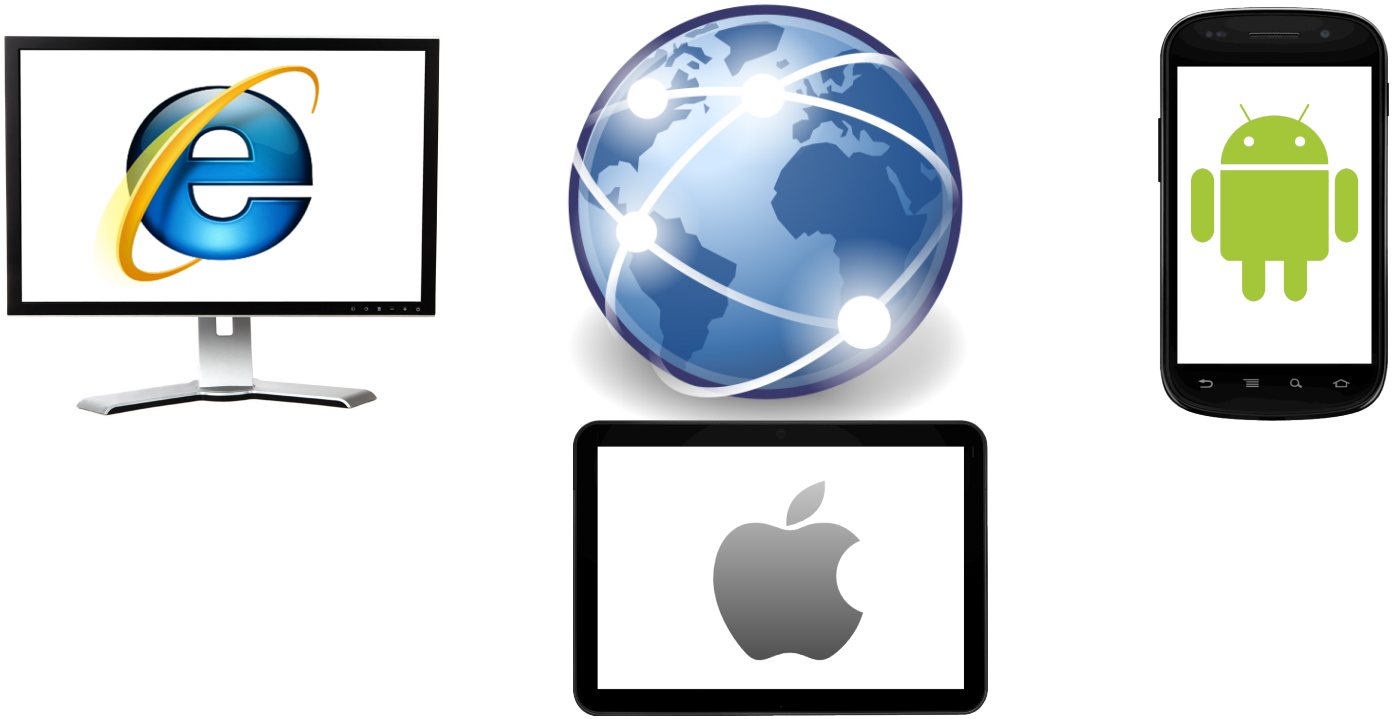
A View To A Kill

WebView Exploitation

Matthias Neugschwandtner
Martina Lindorfer
Christian Platzer

International Secure Systems Lab
Vienna University of Technology

Web - Views



- Consumption of web content shifts to mobile devices
- Typically not through browser but standalone app

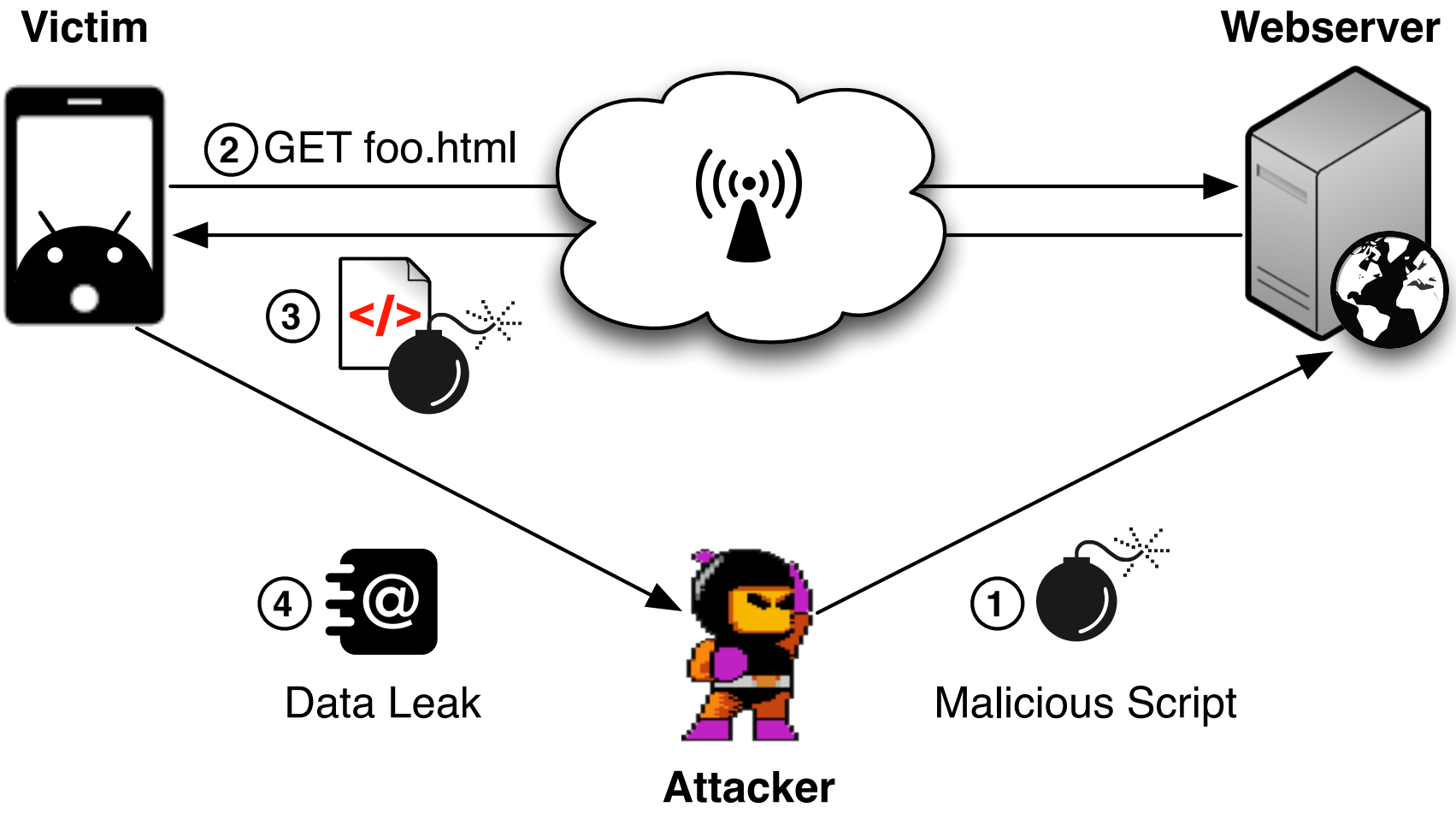
WebView Library

- Browser library for mobile devices
- Available on all popular Smartphone OS
- Allows quick development of web-based apps
  - HTML, JavaScript, CSS
  - Also targeted at inexperienced developers
  - Third party frameworks (Apache Cordova) require no native code at all
  - Updates just require change of web content

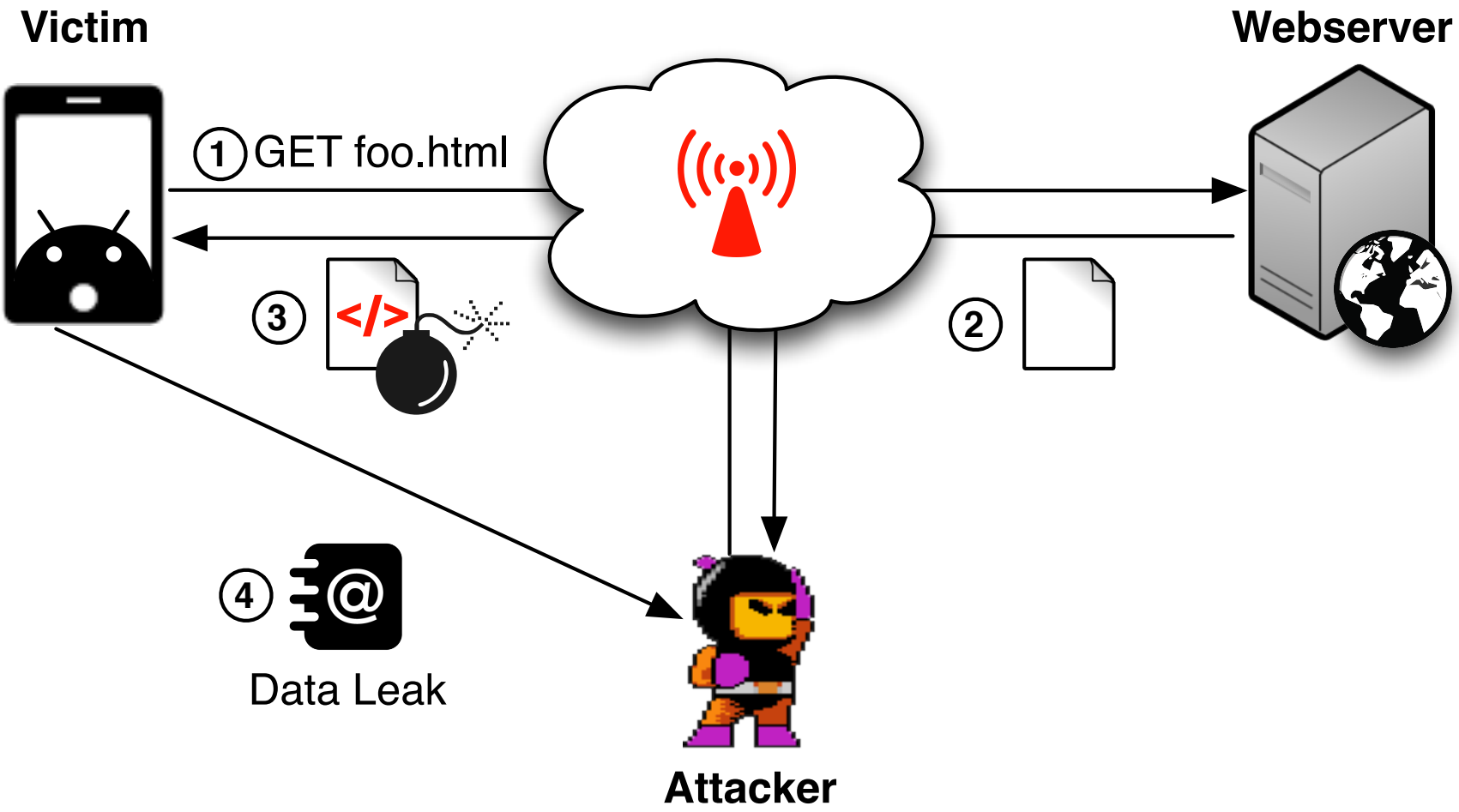
WebView vs. Browser

- Provides access to device functionality via JavaScript
  - Hardware buttons
  - Persistent storage
  - Contacts
  - SMS
  - Location
  - …
- Allows development of more streamlined and capable apps
- No containment of web content (sandbox)

Threat Scenario
Server Compromise



Threat Scenario
Traffic Compromise

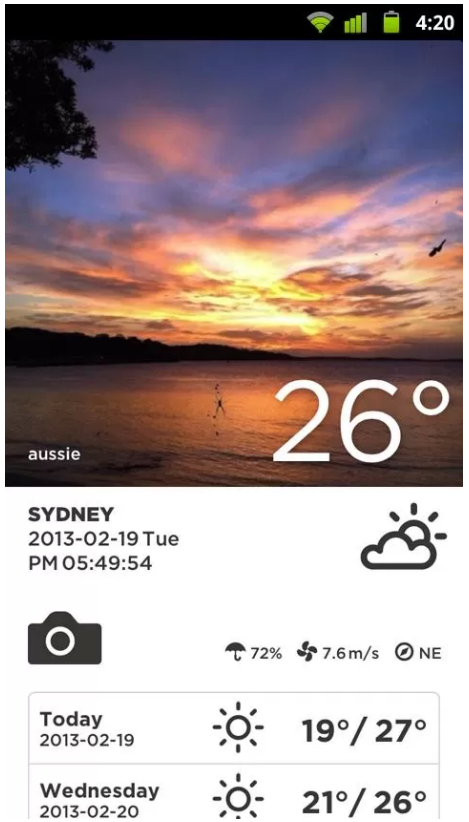


Threat Scenario Comparison

	Server Compromise	Traffic Compromise
Attack leverage	Large (all installations of a single app are affected)	Smaller (depends on number and location of rogue AP)
Encryption	Server takes care of encryption	Only possible with apps that use plain text or don't handle encryption properly
Feasibility	Server dependent	Traffic dependent

Case Study
"Take Weather"

- Social weather-photo sharing app
- Available for iOS and Android
  - 10,000-50,000 installs on Android
- Uses plain HTTP
- Based on Cordova
  - Cross-platform access to contacts, call log, location (GPS)
  - Android: full access to Java



WebView on Android

- Provides JavaScript-Java bridge
  - Expose complete Java objects via
    `WebView.setJavascriptEnabled()`
    `WebView.addJavascriptInterface(<object>, <js_object_name>)`
  - Use reflection to create objects & invoke methods
- Requires signed certificate for HTTPS

Case Study "Jiepang"

- Chinese "Foursquare" – location based social app
- 100,000-500,000 installs
- Permissions to
  - access external storage
  - install packages
- Uses HTTPS, but
  - overwrites default SSL error handler
  - accepts any certificate

Large Scale Evaluation
WebView Prevalence

- 287,512 Android apps submitted to Andrubis
- July 2012 to March 2013
- WebView usage:

WebView related method call	Samples	Percentage
loadURL	166,751	55%
setJavaScriptEnabled	158,042	58%
addJavaScriptInterface	**87,079**	**30%**

Large Scale Evaluation
Traffic Attack Leverage

Traffic Type	Samples	Percentage of JS-enabled samples
Unencrypted HTML or JavaScript	23,048	27%
Lax SSL handling	6,208	7%

Permissions	Samples	Percentage of vulnerable samples
SMS (receive, read, write, send)	3,124	11%
Installation (write, install)	16,726	60%
Privacy (contacts, location)	21,197	76%

Mitigation & Conclusion

- Use of HTTPS and correct certificate handling
  - Signed certificates
  - Certificate pinning
  - WebView targeted at inexperienced developers
- Android 4.2 introduced @JavascriptInterface annotation
  - Will take time until 4.2 is run by a majority of the devices
  - New annotation only prevents reflection attacks
  - Intended functionality is still available