

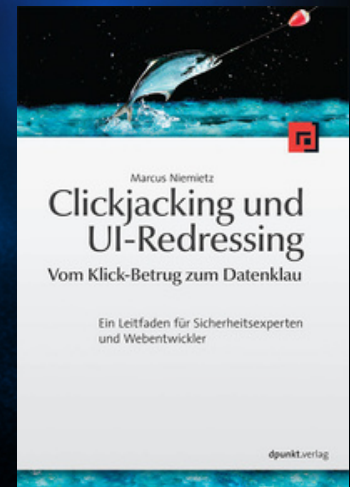


UI Redressing Attacks on Android Devices Revisited

Marcus Niemietz
mail@mniemietz.de

ABOUT ME

- Horst Görtz Institute for IT security
- 3curity GmbH: Trainings, Pentests
- German book about UI redressing
- Speaker at Black Hat, BlueHat, PHDays, Zeronights, OWASP, ...
- Twitter: @mniemietz
- mail@mniemietz.de



CONTENT

1. Introduction
2. UI redressing: Public knowledge
3. UI redressing: Porting to Android
4. New browserless attacks
5. Mitigation techniques against Tapjacking
6. Conclusion and outlook

1. Introduction

INTRODUCTION

- July 1999: German broadband connections for private customers via Deutsche Telekom
 - Decreased the number of phone-based malware
- End of 2009: Dialers are back
New target: Mobile phones

INTRODUCTION

- “Attacks” on mobile phones via
 - Trojan horses
 - Applications with the permission to do phone calls

INTRODUCTION

- Example: WhatsApp Messenger
 - GPS/Network-based location
 - Send SMS/MMS messages
 - Directly call phone numbers
 - Record audio using the microphone
 - Read contact data
 - ...



INTRODUCTION

Is an application **without** phone call permissions able to do a phone call?

INTRODUCTION



2. UI redressing: Public knowledge

UI REDRESSING

- Known since 2002
- Overlooked problem until 2008 → Clickjacking
- Clickjacking \subset UI redressing
 - Subset with attacks like Cursorjacking, Filejacking, Tabnabbing and Tapjacking
 - In general:
The victim has to use a Web browser

UI REDRESSING

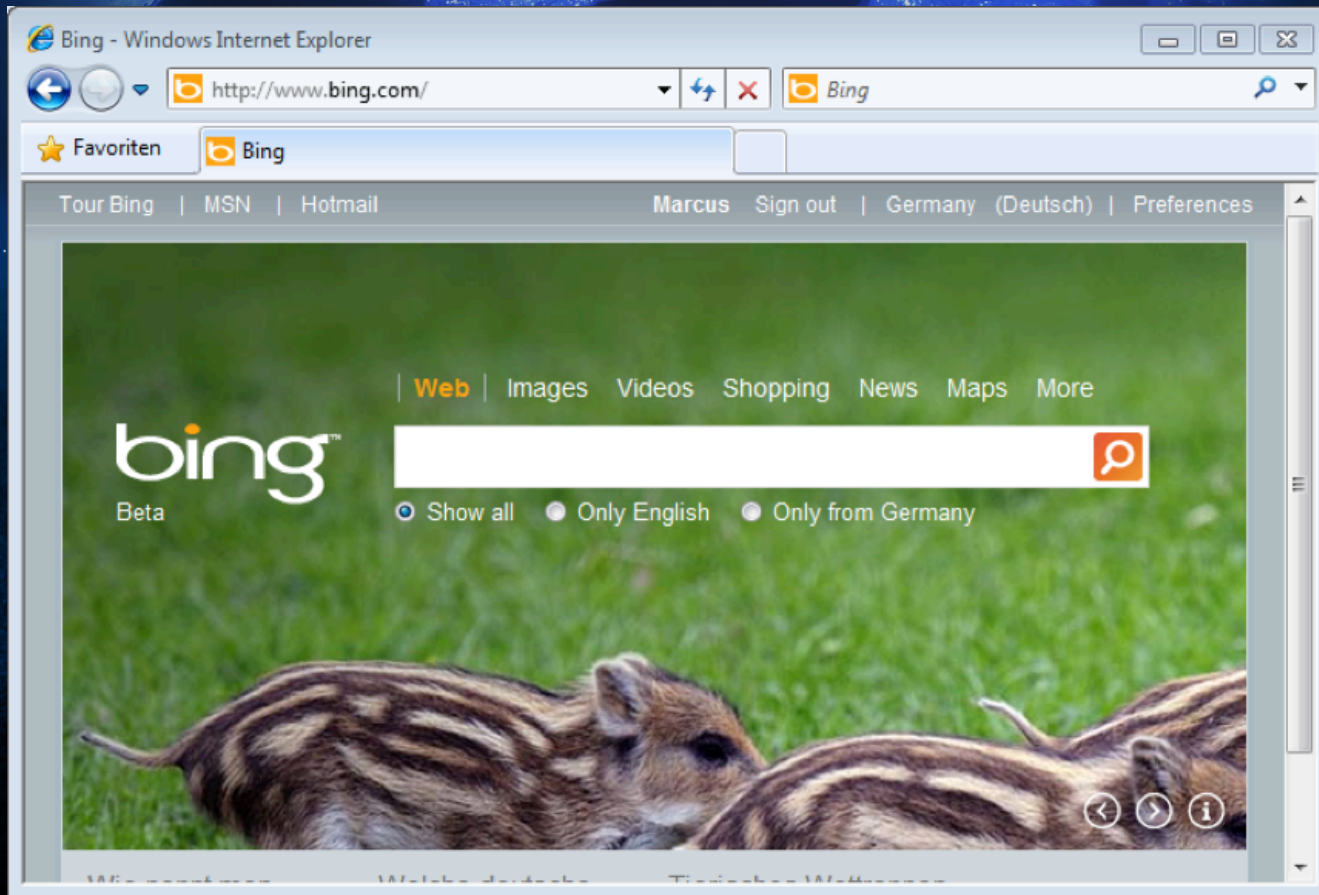
Score: 0 Time: 00:00



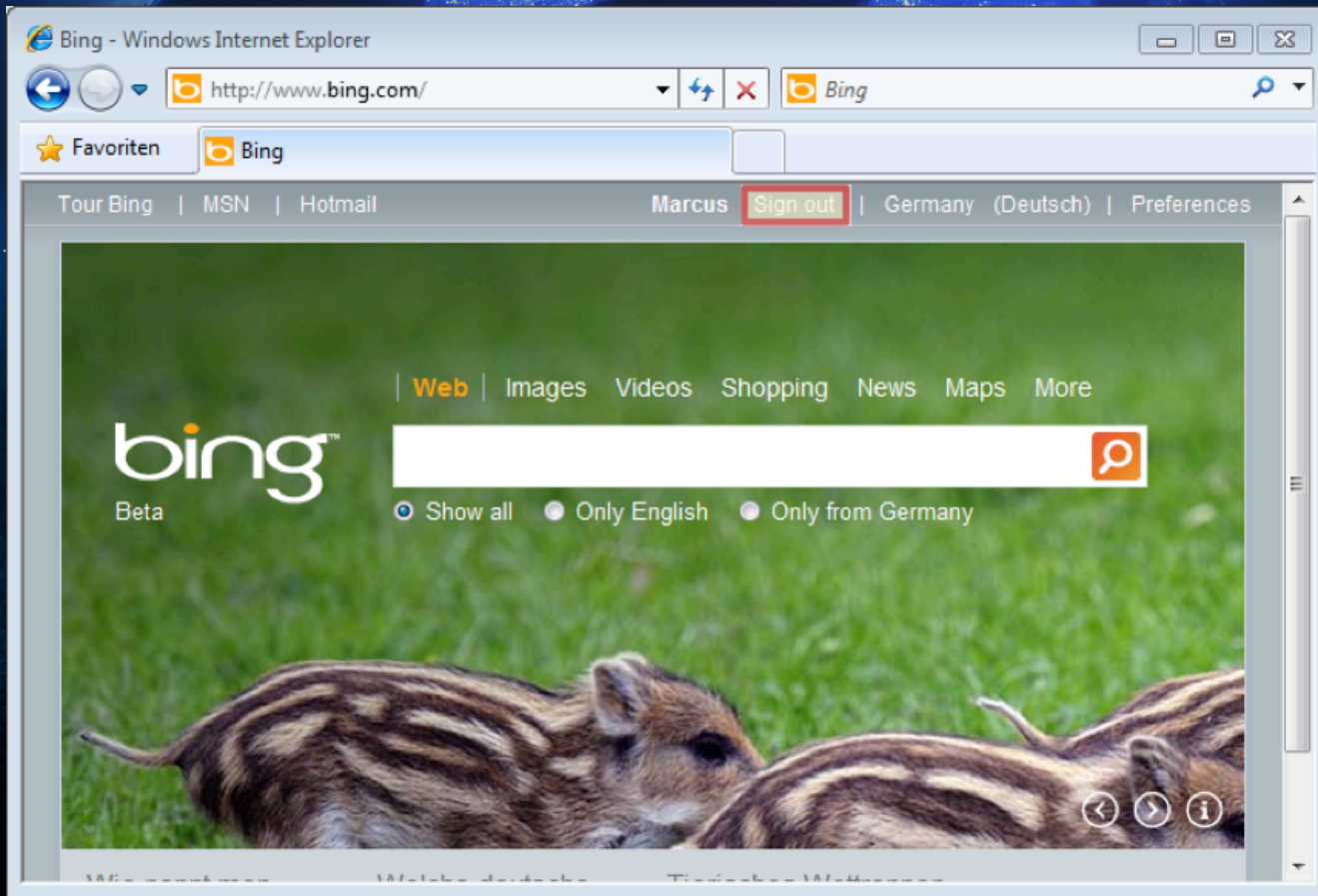
Camera ClickJacking - The Game

START

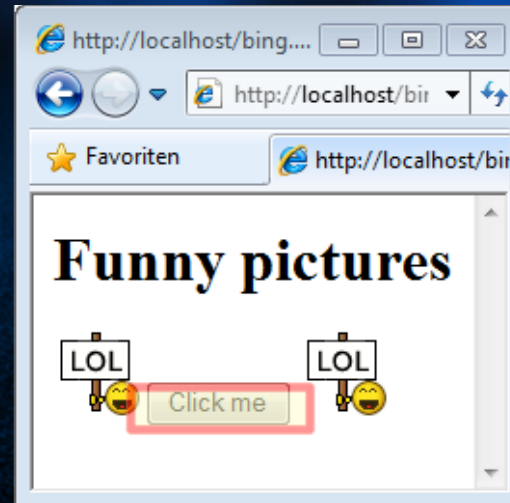
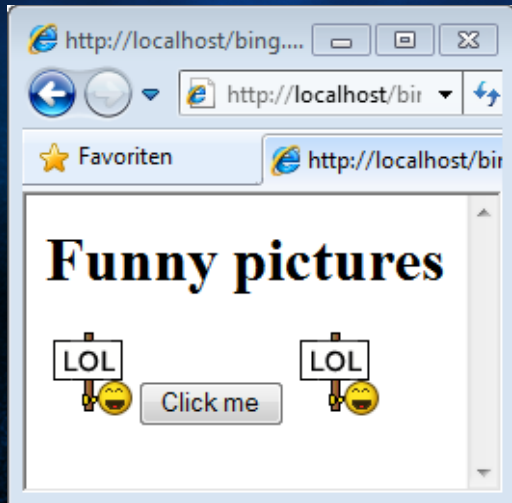
UI REDRESSING



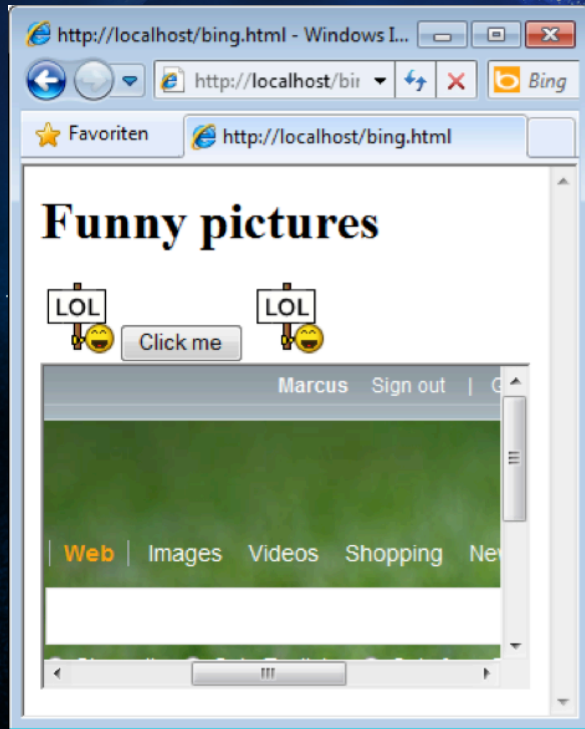
UI REDRESSING



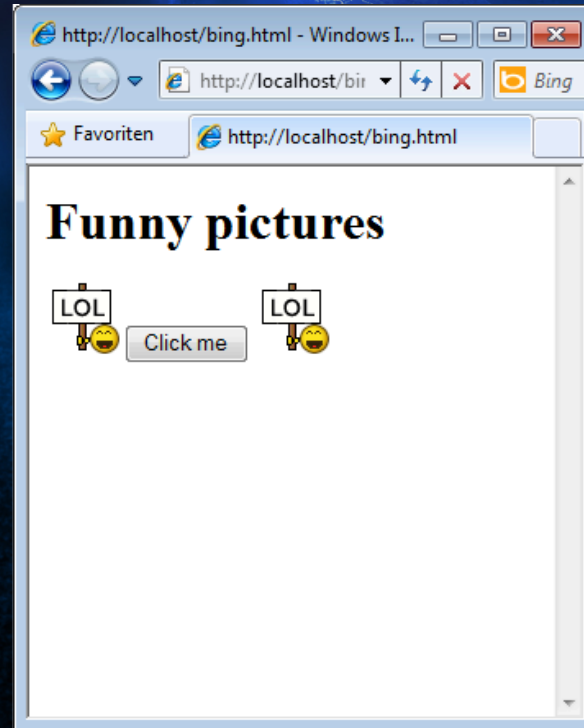
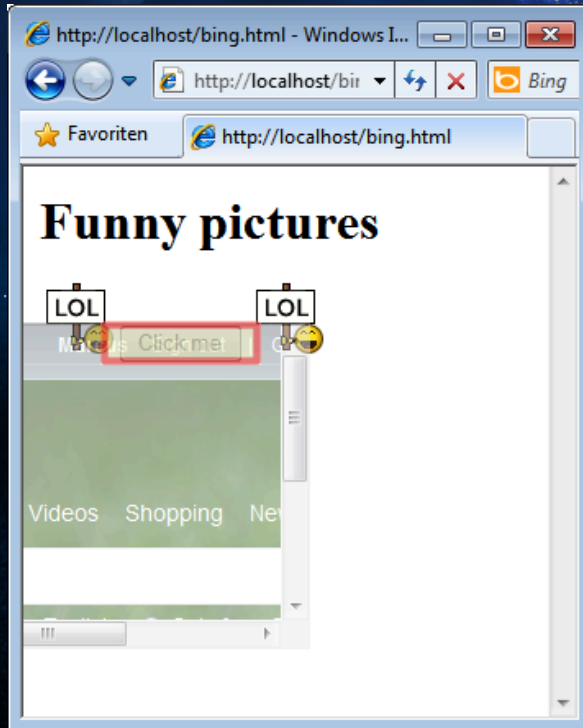
UI REDRESSING



UI REDRESSING



UI REDRESSING



UI REDRESSING

- Clickjacking
- Strokejacking
- Drag-and-Drop
- Content extraction
- Event-recycling
- SVG maskings

UI REDRESSING

- Classic Clickjacking
- Likejacking and Sharejacking
- Nested Clickjacking, Double Clickjacking
- Cookiejacking, Filejacking
- Eventjacking, Classjacking
- Cursorjacking, Tabnabbing
- Combinations with CSRF, XSS, and CSS

UI REDRESSING

■ Countermeasures

- Frame buster
- HTTP header
 - X-Frame-Options
 - CSP
- NoScript

The screenshot shows a security warning dialog box from Mozilla Firefox. The title is "Mozilla Firefox Multiple Vulnerabilities". It lists a security advisory ID "SA39240" with dates "2010-03-31" and "2010-04-05". It indicates "1,251 view" and "0 comments". The severity is "Highly critical". The warning states "Security Bypass" and "System access" and "From remote". The dialog offers several options: "About NoScript...", "Options...", "Allow Scripts Globally (dangerous)", "Allow all this page", "Temporarily allow all this page", "Untrusted", and "Allow secunia.com" (with a sub-option "Temporarily allow secunia.com"). A red "X" icon is visible in the bottom right corner of the dialog.

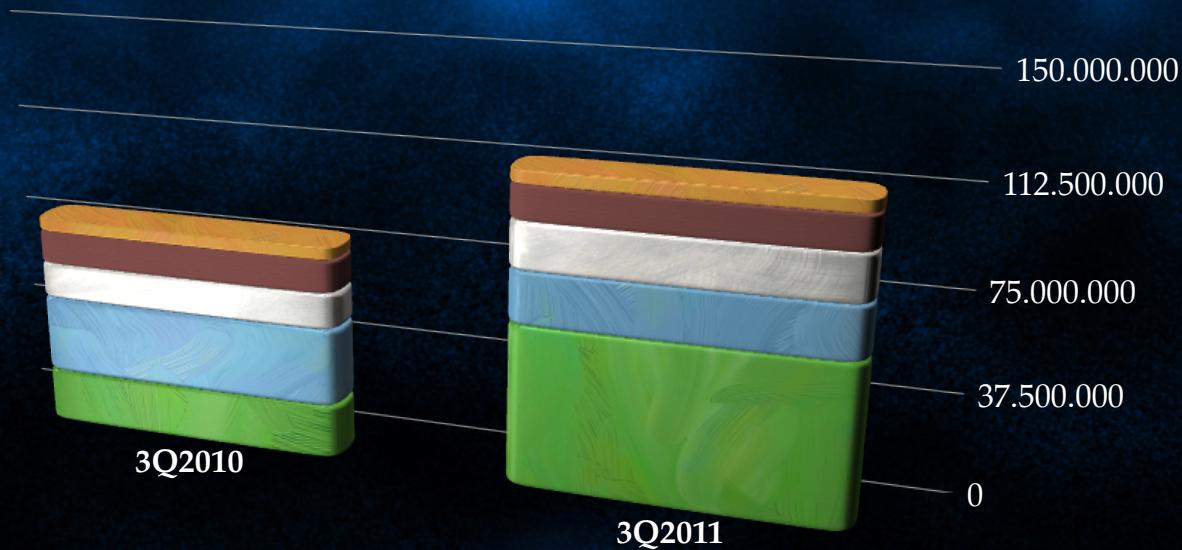
3. UI redressing: Porting to Android

UI REDRESSING

■ Android

- By Gartner (November 2011)
- 2012: Android 66,4%, iOS 19,1% – 2013: Android 78,4%, iOS 15,6%

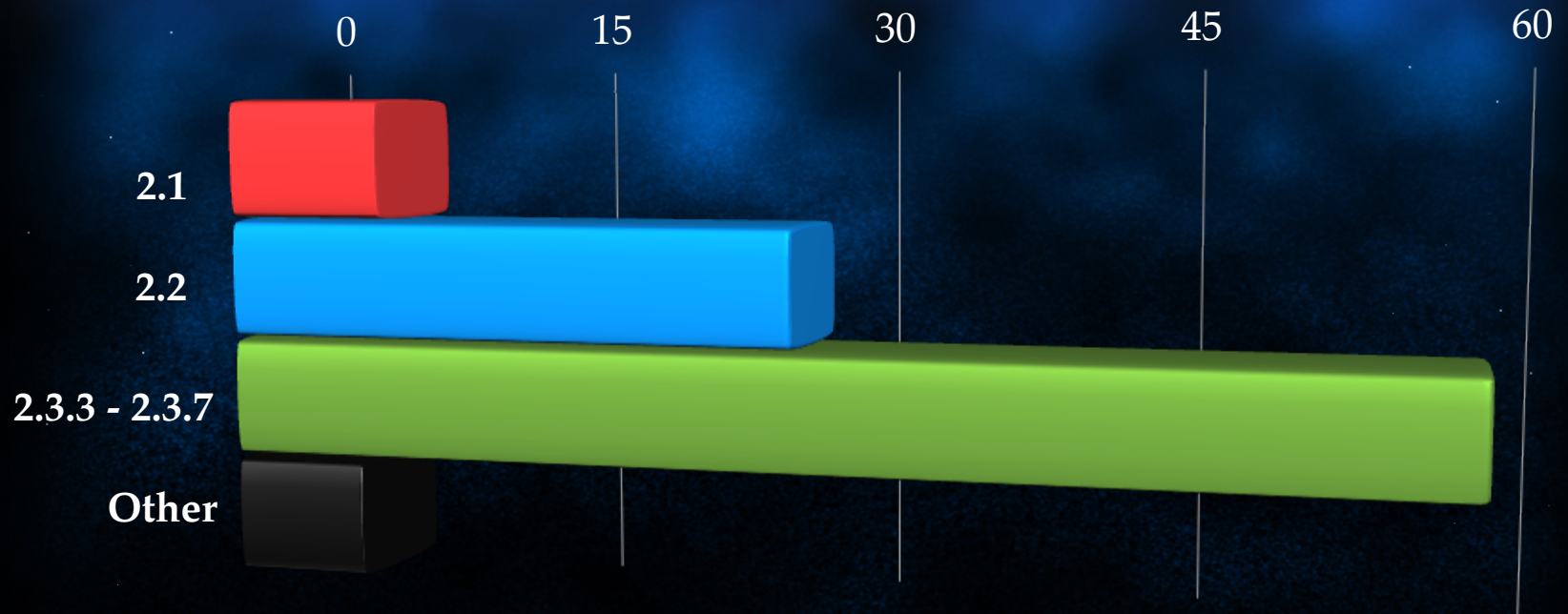
■ Android ■ Symbian ■ iOS ■ RIM ■ Others



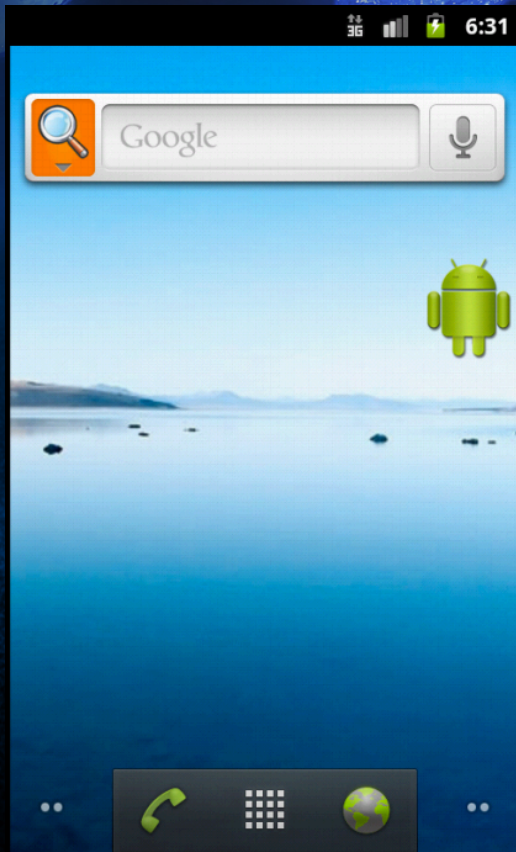
UI REDRESSING

■ Android

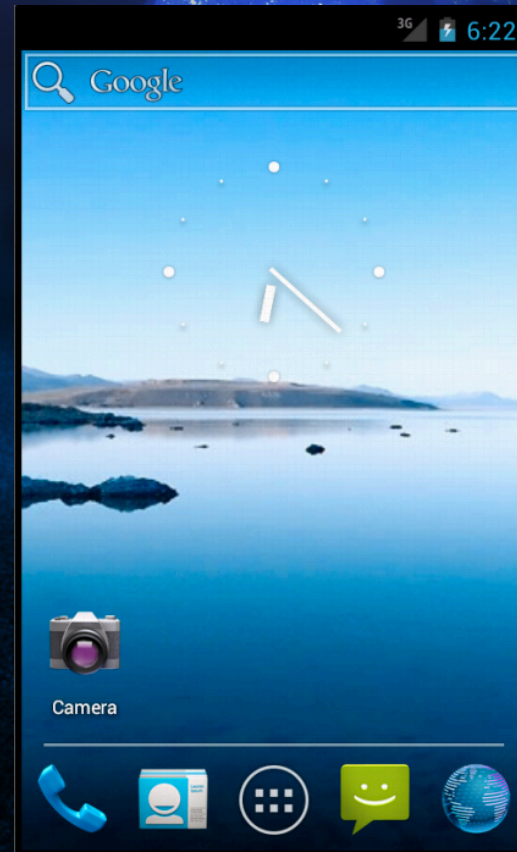
■ Android.com; 14 days; 1 Feb. 2012



UI REDRESSING



ANDROID 2.3.3

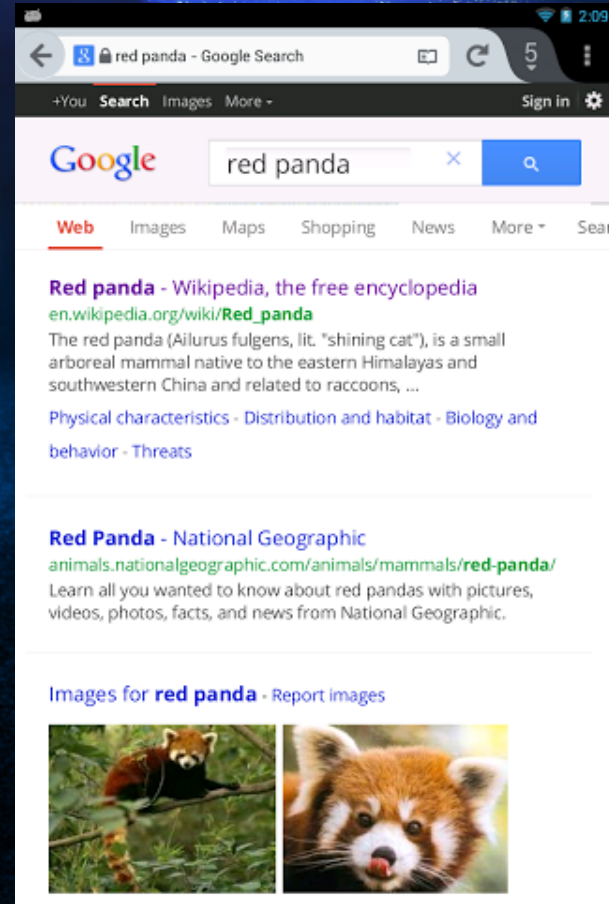


ANDROID 4.0

UI REDRESSING



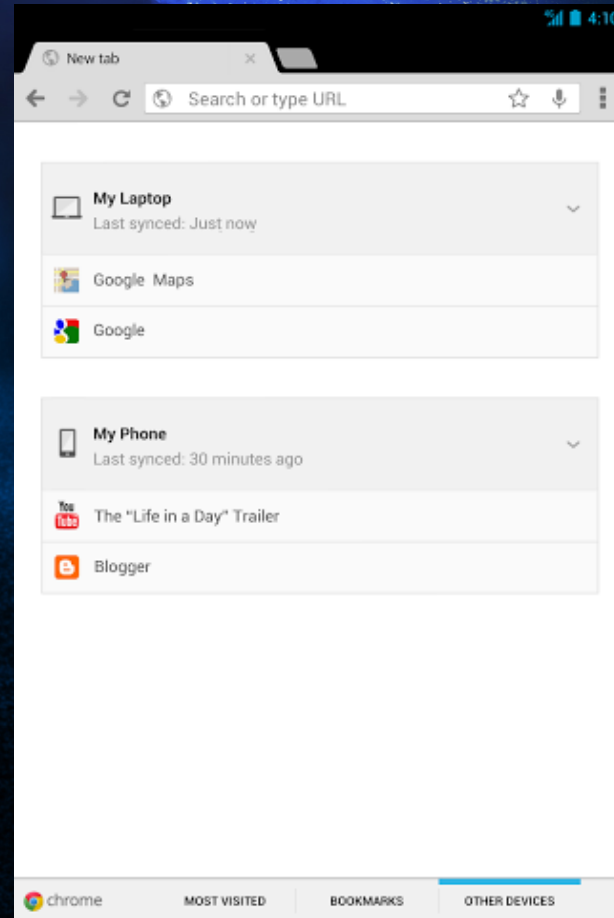
Firefox



UI-REDRESSING



Chrome



UI-REDRESSING



Opera

A screenshot of a mobile website displayed on an Opera browser. The browser's address bar shows the URL 'www.rox.co.uk/selectItem/Diamond_Jewellery/Rings/Solitaire_Rin'. The page content includes a product image of a diamond ring, a detailed description, price information (£4995), and navigation options like 'ADD TO BAG' and 'WISHLIST'. Below the product details, there are sections for 'CUSTOMERS WHO BOUGHT THIS ALSO LIKED...' and 'RECENTLY VIEWED...', each featuring a carousel of similar diamond ring products. The browser's status bar at the top right shows the time as 08:37.

UI REDRESSING



Safari



~~Internet Explorer~~

UI REDRESSING

- Classic Clickjacking, Classjacking, Strokejacking

- Browsers only have to support

- Frames

- CSS

- JavaScript



UI REDRESSING

- Nested Clickjacking, Filejacking, Tabnabbing, Content extraction, Event-recycling and SVG maskings
- Browser-specific features required (e.g. HTML5 attributes)

UI REDRESSING

- Non-transferable attacks

- Cursorjacking



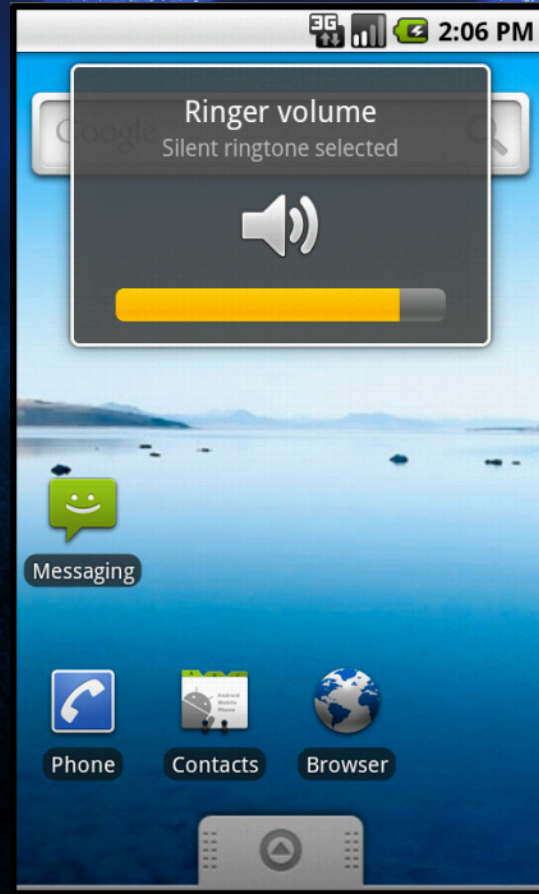
- Double Clickjacking

- Cookiejacking



4. New browserless attacks

NEW BROWSERLESS ATTACKS



NEW BROWSERLESS ATTACKS

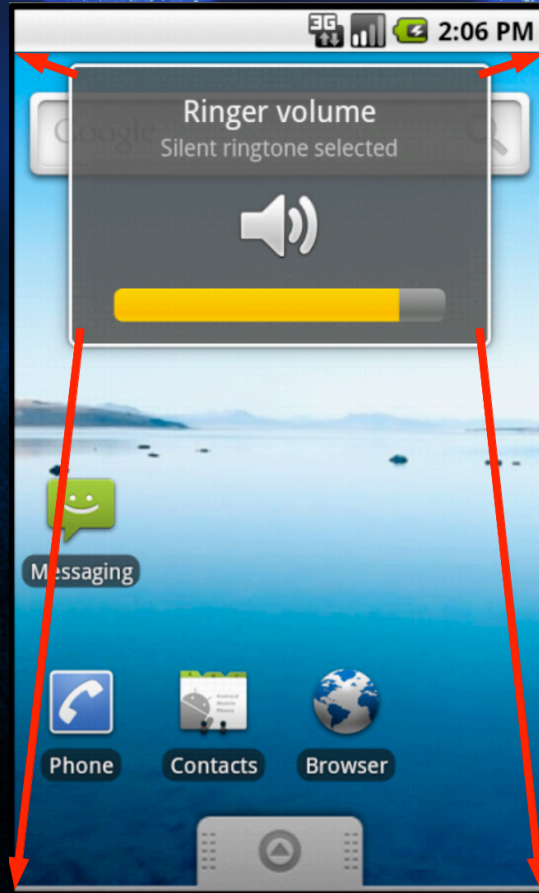
■ Crucial point

- An application can open another application
- A touch gesture on such a message or notification will be passed through to the underlying application
 - Similar to Clickjacking

■ Idea

- Create a notification message, which looks like a normal application

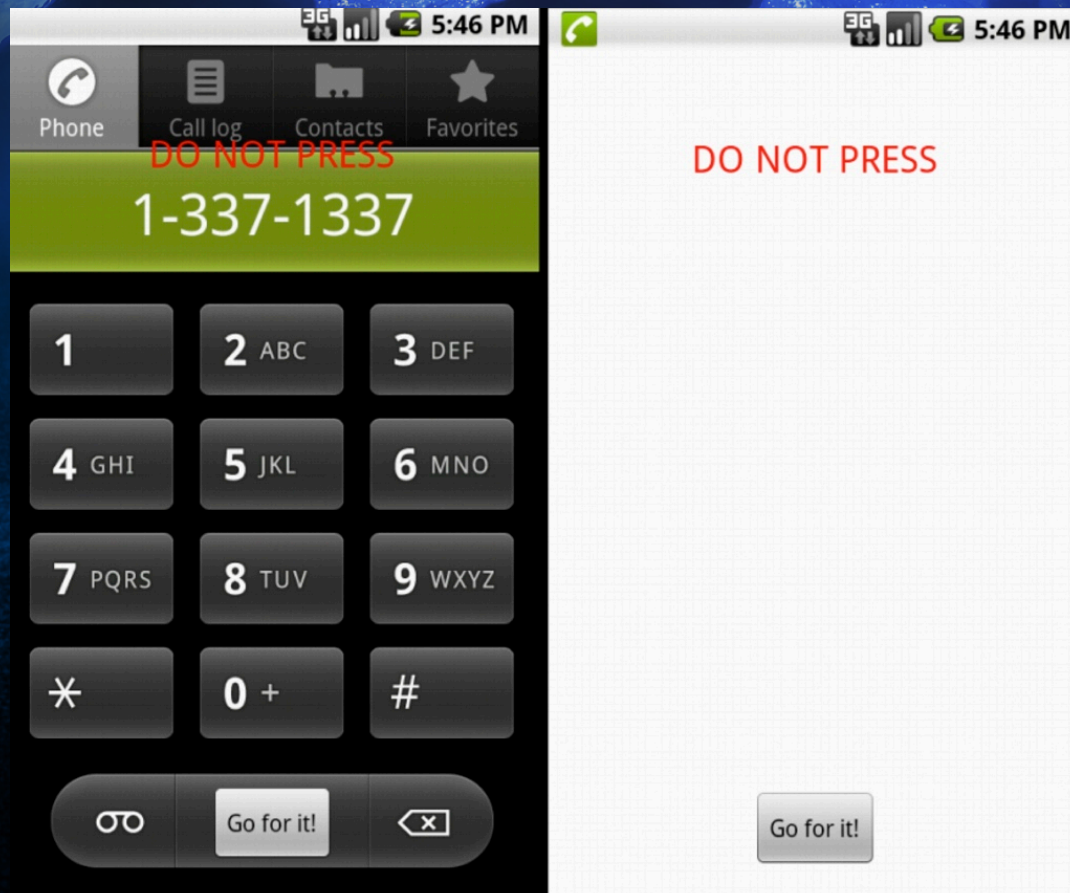
NEW BROWSERLESS ATTACKS



NEW BROWSERLESS ATTACKS



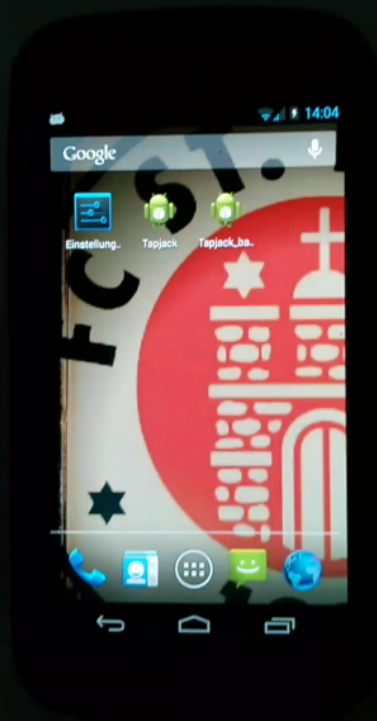
NEW BROWSERLESS ATTACKS



NEW BROWSERLESS ATTACKS

- What an attacker can do
 - Contact data manipulation
 - Native browser utilization
 - Touch gestures logging
 - Predefined phone calls
 - Installing applications in the background ~~FIXED~~

UI REDRESSING



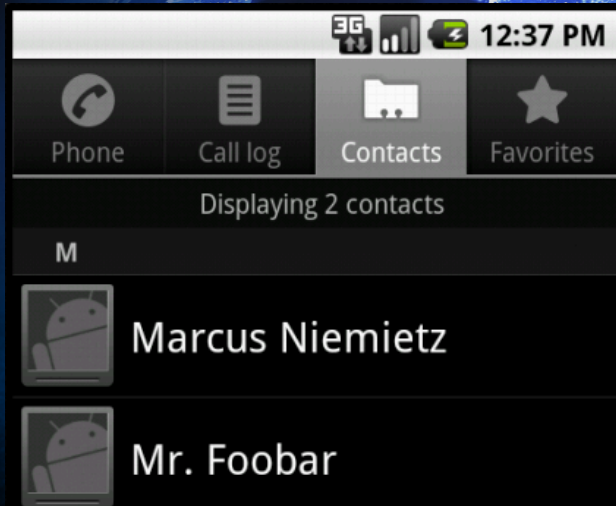
NEW BROWSERLESS ATTACKS

- All of these attacks are using the same technique
 1. There is a visible attacker's application in form of a notification in the foreground
 2. There is a target application in the background

NEW BROWSERLESS ATTACKS

- There is a limited number of operations like opening the phone call application
- Solution: Unauthorized home screen navigation attack
 - Substantially extend the limited set of attacks
 - An attacker needs more touch gestures of a victim

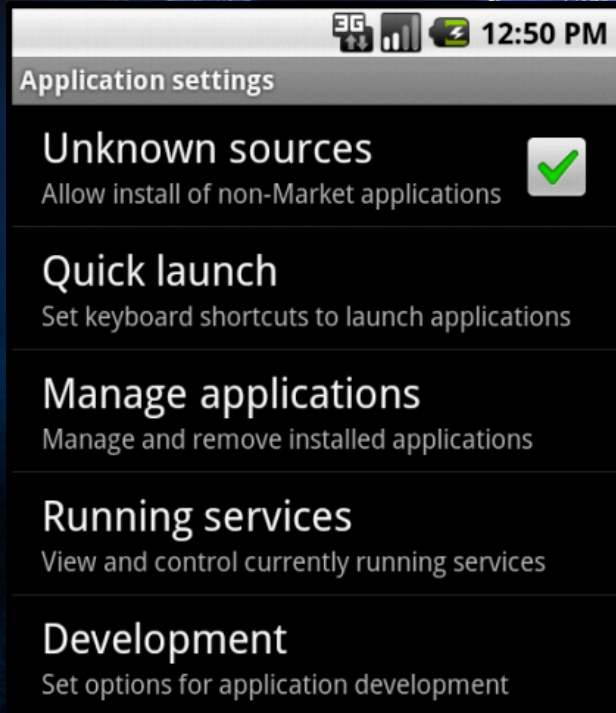
NEW BROWSERLESS ATTACKS



Manipulate contact data

Delete contact data:
4 touch gestures

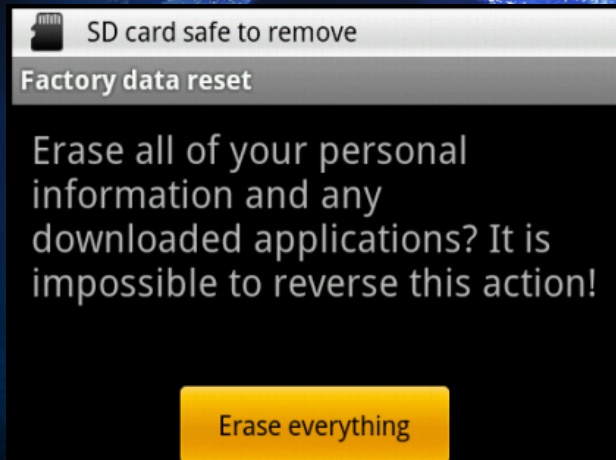
NEW BROWSERLESS ATTACKS



Application settings

Allow the installation of
non-Market applications:
5 touch gestures

NEW BROWSERLESS ATTACKS



Wipe via „Factory Reset“

Just **6** touch gestures



5. Mitigation techniques against Tapjacking

MITIGATION TECHNIQUES

- Android touch filter
 - Blocks touch gestures received whenever view's window is obscured
 - `setFilterTouchesWhenObscured()` or, alternatively, with the attribute `android:filterTouchesWhenObscured`
- Not enabled by default and they are only available in Android versions higher than 2.2

6. Conclusion and outlook

CONCLUSION AND OUTLOOK

- Most of the existing UI redressing attacks can be used with very little effort
- An attacker can use UI redressing with and without a browser on a mobile phone
- Important: Dialers are back
- There are countermeasures like XFO, CSP, and `setFilterTouchesWhenObscured()`

Thank you for your attention.

Any questions?