How Android's UI Security is Undermined by Accessibility

Anatoli Kalysch (**Speaker**), Davide Bove, and Tilo Müller

November 30, 2018

Friedrich-Alexander-Universität Erlangen-Nürnberg
Department of Computer Science
IT Security Infrastructures Lab
Software Security Research Group

DEEPSEC

FAU FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

whoami

- PhD Student at FAU
- Research Focus on Mobile Security and Program Analysis
  - Malware Analysis
  - App Analysis Environments
  - UI Security
  - App Reverse Engineering
- Also tutoring courses in Reverse Engineering and IT Forensics

Outline

Outline

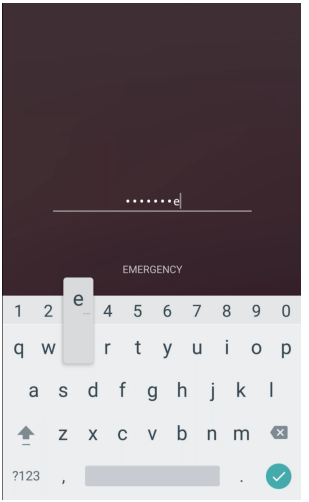
UI Security

- Android features strong app separation concepts, e.g.,
  - Sandboxing
  - Binder-assisted window management
  - Permission-based concepts
- UI offers a user-centered perspective on data and apps
- Flaws in the UI allow to mitigate these separation concepts and leak data or overtake the UI [3, 2].

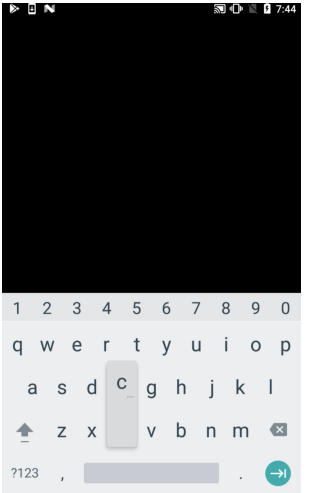
UI Security in an A11y Context

- A11y services have a major impact on UI security due to them
  - being notified about *every* UI change, and
  - being able to take action for the user
- Actions can be automated and performed a lot faster
  - e.g., granting permissions and generating user input [2],
  - starting screen recordings or even
  - enabling a new IME and registering it as the new default
- Since many security measures rely on user confirmations this creates immense potential for abuse

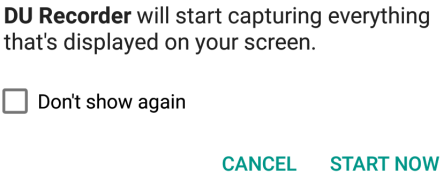
Secure Flag


••••••••e
EMERGENCY
e
1 2 3 4 5 6 7 8 9 0
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
?123 ,
.

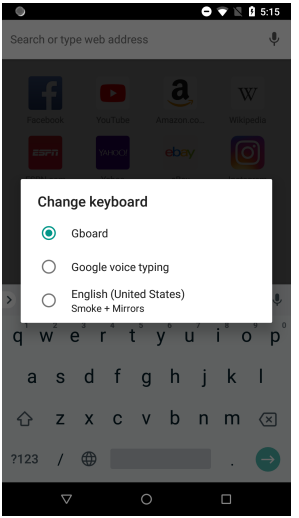
Secure Flag



User Dialog for Screen Recordings



Register your own IME



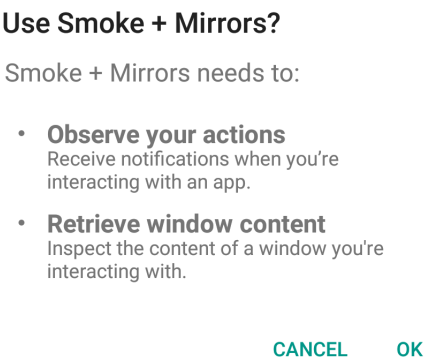
A11y TextView Sniffing

- A11y services are notified about UI changes, e.g., when the user generates input such as text
- Every input character generates several a11y events which can be problematic for password / sensitive information fields
- Which is why all contents of password fields are censored with the ‘dot’ character to not leak any information ...
- ... except the last character which is visible for 2.5 seconds after entry!

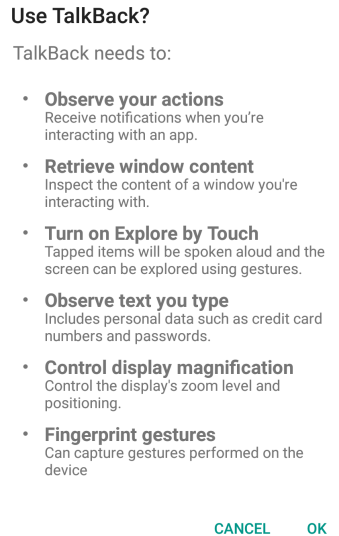
Missing Settings Synchronization

- Using Unstructured Supplementary Service Data (USSD) codes allows you to change phone, network, and carrier options
- Some of these options are hidden to the normal user others can be interacted with from the settings app
- Changing options through USSD codes does not affect the values displayed in the setting app
- An a11y service can actually interact with settings apps and input USSD codes through the phone application

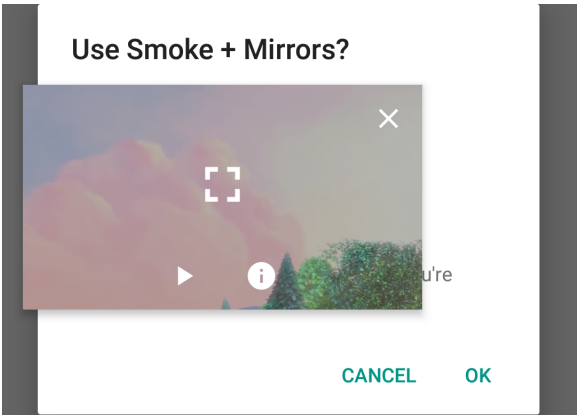
Misleading Capabilities



Misleading Capabilities



Picture-in-Picture Mode



Outline

Distribution of A11y Services in App Stores

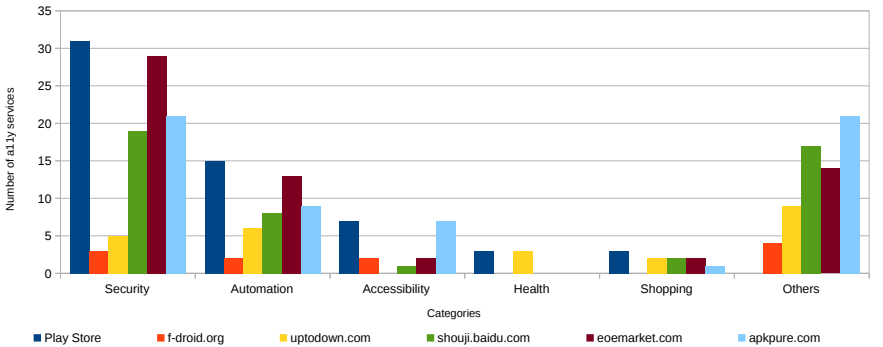


Figure: Number of a11y services per app category.

App Logins Under Attack

Category	Number of apps with a login	Percentage of logins vulnerable against		
		A11y Events	*Screen Records*	*Malicious IMEs*
Business	116	100%	100%	100%
Communication	47	100%	100%	100%
Dating	63	100%	100%	100%
Entertainment	58	100%	100%	100%
Finance	172	84.9%	96.5%	94.2%
Games	104	95.2%	100%	100%
Health	57	98.3%	100%	100%
Shopping	42	95.2%	100%	100%
Social	99	100%	100%	100%
Travel	45	97.7%	100%	100%
Summary	803	95.6%	99.3%	98.8%

Table: Out of 1100 apps 803 had a login screen, most of them being vulnerable.

App Protection Measures

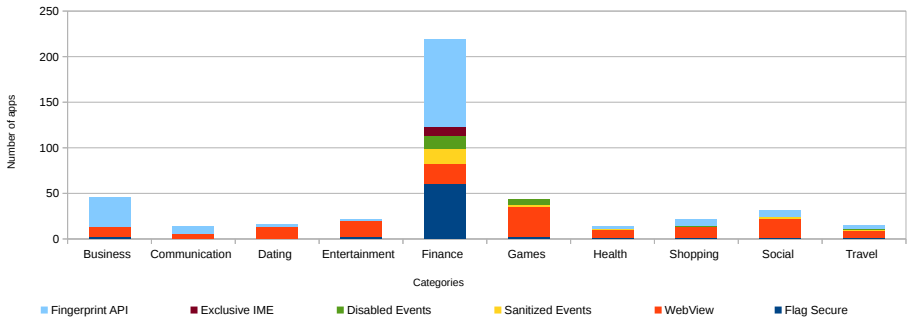


Figure: Employed security mechanisms per category.

Countermeasures

- A11y event filtering or sanitization
- Behavioral Listeners
- Window Punching
- In-App Keyboards
- Fingerprint API

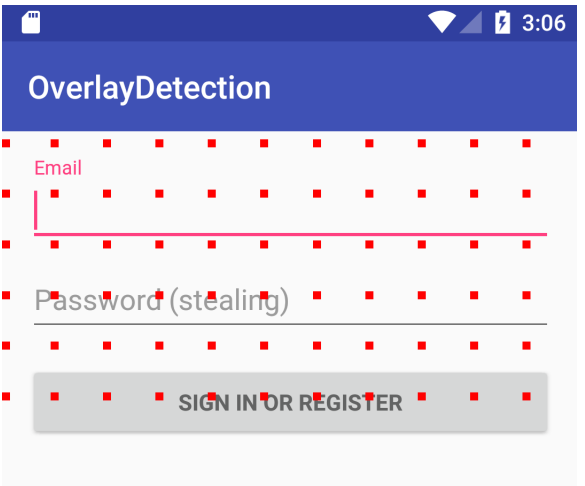
A11y Event Filtering

- public boolean dispatchPopulateAccessibilityEvent(AccessibilityEvent event)
- public void onPopulateAccessibilityEvent(AccessibilityEvent event)
- public void onInitializeAccessibilityEvent(AccessibilityEvent event)
- public void onInitializeAccessibilityNodeInfo(AccessibilityNodeInfo accessibilityNodeInfo)

Behavioral Listeners

- We can abuse the limitations of a11y services as a countermeasure
- This however excludes people using them from using the app
- A11y services can only simulate ‘click’ events
  - No TouchDown or TouchUp event generated
  - Switching to the corresponding Listeners

Window Punching



Vetting Attacks against Countermeasures

Attack	Vulnerable Android Versions					Possible Countermeasures
	6.0	7.0	7.1.2	8.0	8.1	
A11y Event Sniffing	✗	✓	✓	✓	✓	a11y event sanitizing, fingerprint auth.
A11y Screen Recording	✓	✓	✓	✓	✓	secure flag *and* in-app keyboard
A11y-enabled Malicious IME	✓	✓	✓	✓	✓	in-app keyboard *and* behavior listeners
A11y-based Ad Hijacking [2]	✓	✓	✓	✓	✓	a11y event sanitizing
Overlay and a11y assisted password stealing [2]	✓	✓	✓	✓	✓	a11y event sanitizing, window punching
Keyboard App Hijacking [2]	(✓)	(✓)	(✓)	✗	✗	in-app keyboard *or* enforcing Gboard update
Full App Passthrough / Clickable Overlays [**?**]	✓	✓	✓	✓*	✓*	window punching
Partial App Clickable Overlays [**?**]	✓	✓	✓	✓*	✓*	window punching
Context-aware Clickjacking / Hiding [2]	✓	✓	✓	✓*	✓*	window punching
Keystroke Inference [2]	✓	✓	✓	✗	✗	in-app keyboard *and* window punching

Table: A11y and overlay-based attacks presented here and in previous work by different authors.

Take Aways

- A11y Services can sniff passwords upon activation
  - If they are additionally allowed to take actions for the user even silent IME installs and screen recordings become possible
  - Device tampering is possible as well
- Application developers are not aware of this threat
  - 99.25% of apps with a login on Google Play were vulnerable to credential leakages
  - Currently deployed protection mechanisms do not offer adequate protection
- A11y Services and UI attack scenarios are a viable threat as recent ‘advances’ in malware have shown [4][1]

Disclosure Process

- Vulnerabilities and bugs reported through bug and security reports
  - 7 reports all together
  - Highest rating among them is ‘Low’ most were rated ‘NSBC’ (Non-Security Bulletin Class -> probably won’t get a fix anytime soon)
- Reports were submitted as between December 2017 and March 2018
- Selected app developers from the categories finance and healthcare were notified about the bugs in their login fields in August
- Notification of all developers planned until the end of 2018

Demo

Thank you.

Questions?

Contact

Website: https://www.cs1.tf.fau.de/person/anatoli-kalysch/
Email: anatoli.kalysch@fau.de
Twitter: @aka_kaly

PoC snippets and selected PoC countermeasure projects available (starting December 1st) at:

https://github.com/anatolikalysch/roots_a11y

References

[1] Yair Amit.
Accessibility clickjacking – android malware evolution, 2016.
https://www.symantec.com/connect/blogs/accessibility-clickjacking-android-malware-evolution, accessed on 11. August 2018.

[2] Yanick Fratantonio, Chenxiong Qian, Simon P Chung, and Wenke Lee.
Cloak and dagger: from two permissions to complete control of the ui feedback loop.
In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 1041–1057. IEEE, 2017.

[3] Yeongjin Jang, Chengyu Song, Simon P Chung, Tielei Wang, and Wenke Lee.
A11y attacks: Exploiting accessibility in operating systems.
In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 103–115. ACM, 2014.

[4] Swati Khandelwal.
Ransomware not just encrypts your android but also changes pin lock, 2017.
https://thehackernews.com/2017/10/android-ransomware-pin.html, accessed on 20. August 2018.