Identity Confusion in WebView-based Mobile App-in-app Ecosystems

Lei Zhang [1,⋆], **Zhibo Zhang** [1,⋆], Ancong Liu [1], Yinzhi Cao [2], Xiaohan Zhang [1], Yanjun Chen [1], Yuan Zhang [1], Guangliang Yang [1] Min Yang [1]

1. Fudan University
2. Johns Hopkins University

⋆: The first two authors have contributed equally to this work.

App-in-App Ecosystem

- Super-app
  - A mobile app with rich functionalities, often delegate their functions to other parties (sub-apps)
  - e.g., Paytm, Snapchat, TikTok, WeChat
- Sub-app
  - brings rich content and services
  - native app like experience

source: https://paytm.com

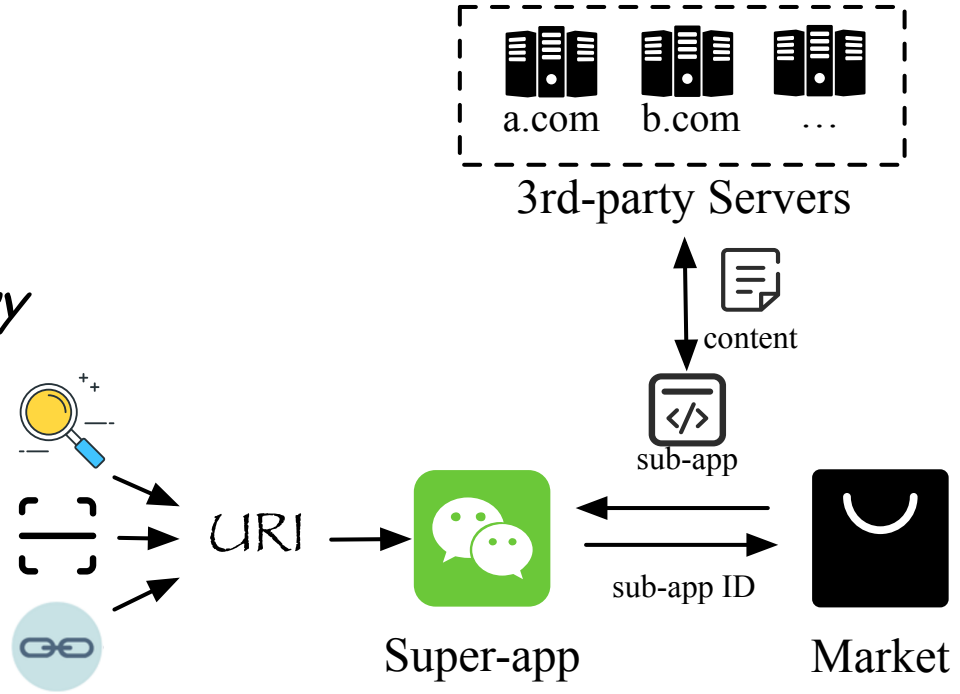
The Popular Trend

- More than 47 high-profile super-apps
  - world-wild
    - Asia (Grab, Line, Paytm), European(VK), and the U.S. (Microsoft Teams)…
  - large user base
    - with 46B+ downloads in total
- Huge amount of sub-apps
  - e.g., 3.8M+ sub-apps in WeChat

source: https://paytm.com

Programming Model & Lifecycle

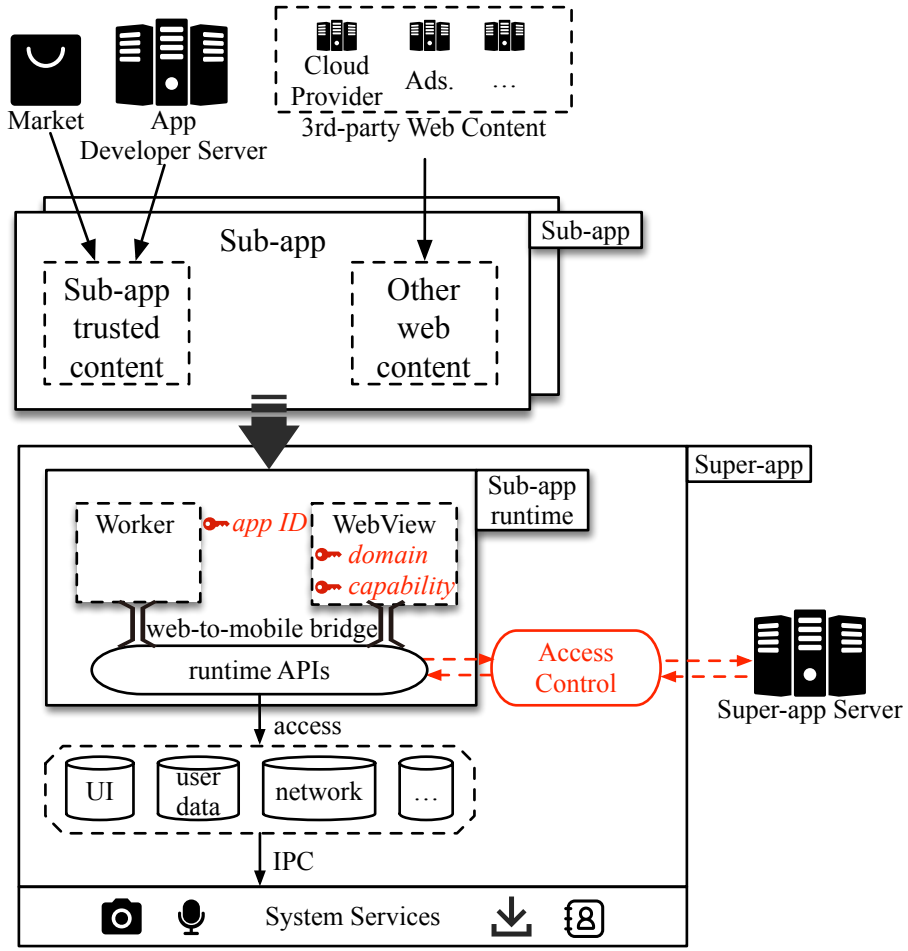
- Program language
  - JavaScript, HTML, and CSS
- Upload to super-app market
  - driven by URI
    - e.g., *super-app://sub-appID/path/query*
- Load multi-party resource
  - from sub-app market
  - from third-party servers
- Access privileged APIs



Crucial Question: determining who can call specific privileged APIs

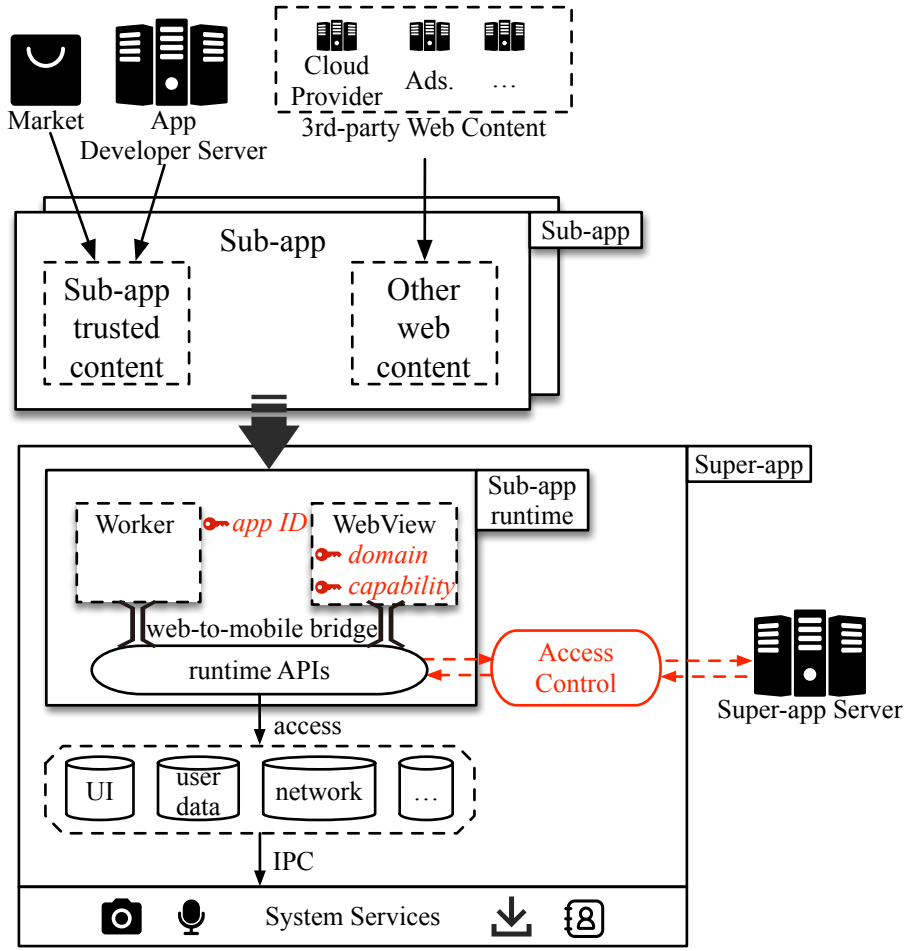
A Survey Study

- Popular Super-app Runtimes
  - an embedded browser instance
    - (customized) WebView in Android
    - WKWebView in iOS
  - web-to-mobile bridge
    - enable JS to call Java functions
  - runtime APIs
    - 50% are un-documented
    - 80% are privileged
      - access user data
        - e.g., account, bank info, phone number…
      - access OS resources
        - e.g., camera, location…



A Survey Study

- Existing Identity Checks
  - Domain Name
    - represents a server and contents delivered from the server
  - App ID
    - assigned by a super-app to the sub-app
  - Capability
    - a secret issued by either a super-app or a server and checked based on exact match.

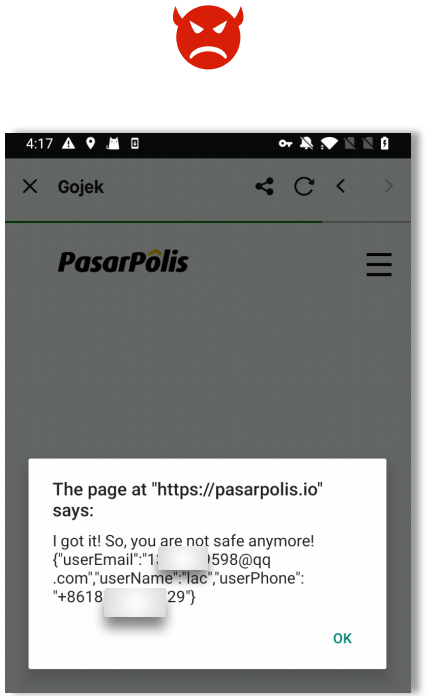


Identity Confusion

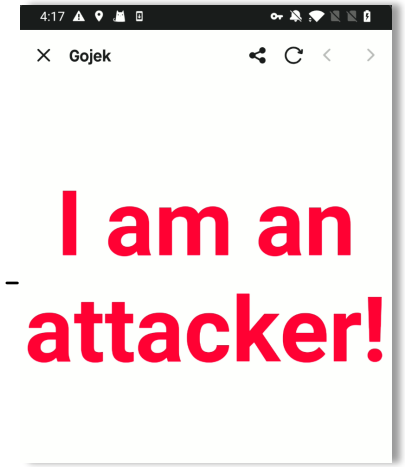
- Definition
  - intended identity can be broader than or different from it actually represents
  - disobey the least privilege principle
- Domain Name Confusion
  - Privileged web domain in an unprivileged sub-app
- App ID Confusion
  - Unprivileged web domains in a privileged sub-app
- Capability Confusion
  - Privileged capability obtained by an unprivileged sub-app or domain

Identity Confusion Attack

- Once exploited, the attacker can
  - inject phishing web page to popular sub-app
  - steal user data (address, payment info, phone number, email…)
  - abuse OS resources (open microphone, install malicious apk)



steal email, phone number

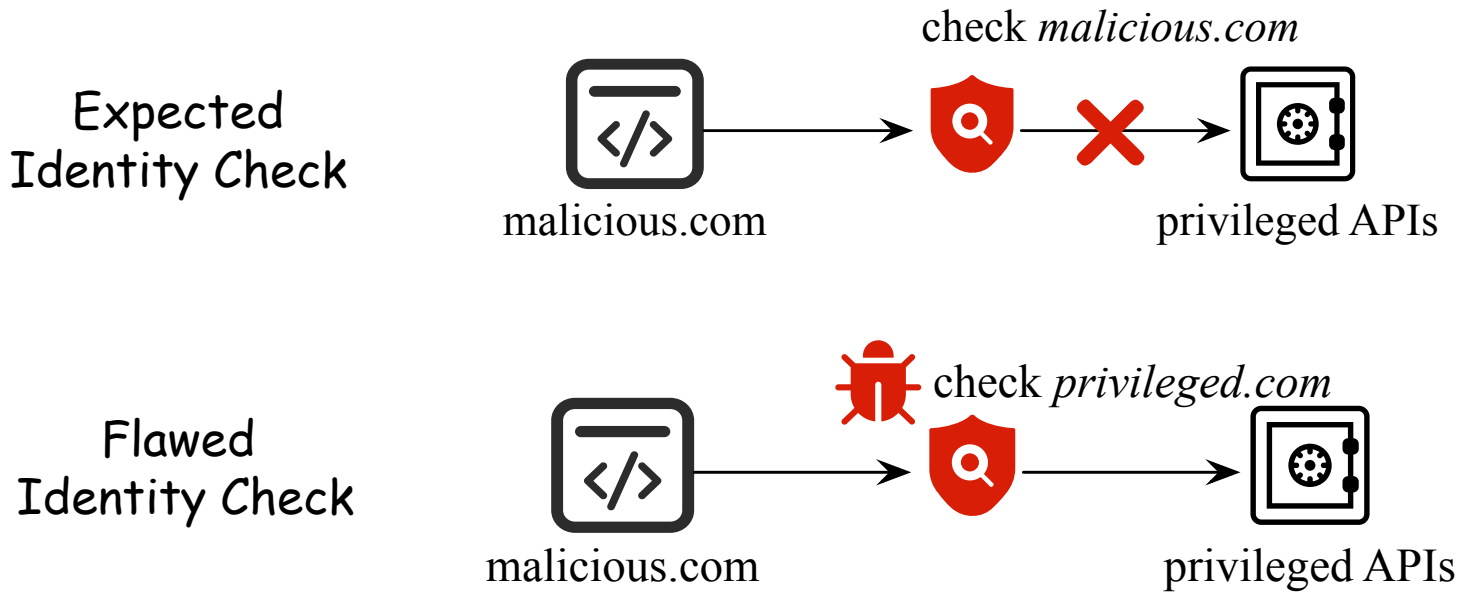


phishing deeplink

steal address, payment info

A Taxonomy Study

- Domain Name Confusion
  - happens in domain based identity check
  - checked domain != actual domain



Domain Name Confusion

- **Type 1: Timing-based Confusion**
  - Case1. super-apps use *onPageStarted()* to get identity
    - race condition between different threads of WebView

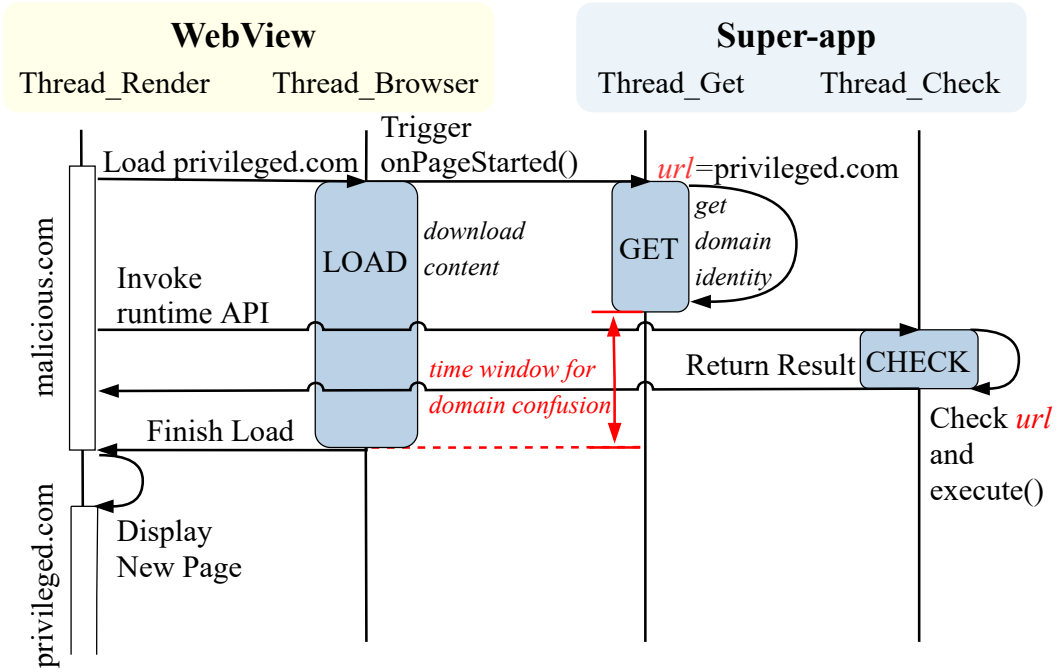


Figure: Race between WebView's Render and Browser Threads

Domain Name Confusion

- **Type 1: Timing-based Confusion**
  - Case2. super-apps use *getUrl()* to get identity
    - race condition between different threads of super-app

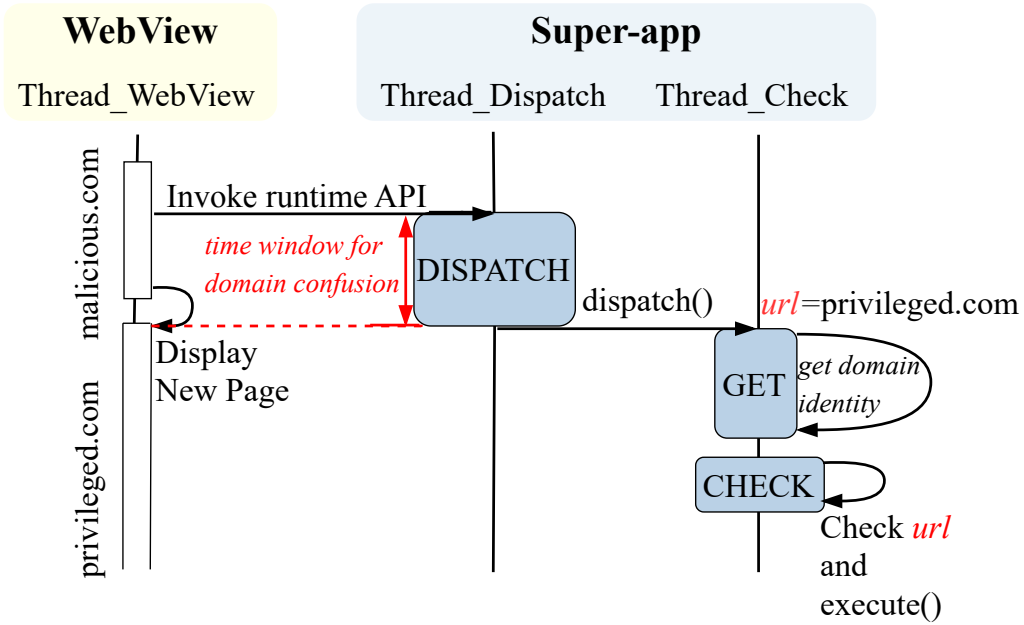


Figure: Race between super-app's Dispatch and Check Threads

Domain Name Confusion

- **Type 2: Frame-based Confusion**
  - an iframe acts on behalf of the top frame's identity

Class Name	Method Signature of Event Handlers	Domain Name Confusion	
		Timing-based	Frame-based
	Getter Method:		
WebView	getOriginalUrl ()	✔	✔
	getUrl ()	✔	✔
	Callback Method:		
WebViewClient	doUpdateVisitedHistory (WebView view, **String url**, boolean isReload)	✔	✔
	onLoadResource (WebView view, **String url**)	✔	
	onPageCommitVisible (WebView view, **String url**)	✔	✔
	onPageFinished (WebView view, **String url**)	✔	✔
	onPageStarted (WebView view, **String url**, Bitmap favicon)	✔	✔
	onReceivedClientCertRequest (WebView view, **ClientCertRequest request**)	✔	✔
	onReceivedError (WebView view, **WebResourceRequest request**, WebResourceError error)	✔	
	onReceivedHttpAuthRequest (WebView view, HttpAuthHandler handler, **String host**, String realm)	✔	
	onReceivedHttpError (WebView view, **WebResourceRequest request**, WebResourceResponse errorResponse)	✔	
	shouldInterceptRequest (WebView view, **WebResourceRequest request**)	✔	
	shouldOverrideUrlLoading (WebView view, **WebResourceRequest request**)	✔	
WebChromeClient	onReceivedTouchIconUrl (WebView view, **String url**, boolean precomposed)	✔	✔

Table 3: The domain name confusion in using WebView's event handlers to obtain identity information. We measure them at time and frame dimensions.

Prevalence & Consequence

- Step II: Vulnerability Analysis
  - Methodology
    - static analysis on super-apps to find whether a vulnerable API is used
    - write test cases and exploits
  - Cross Platform Verification
    - use the Proof of Vulnerability (PoV) for Android versions of super-apps to verify their iOS versions

```
1    //JavaScript
2    window.setInterval(function(){
3    res = nativeInterface.framelessPostMessage('
     {"id":1,"func":"authentication.getAuthToken","
     args":[["privileged.com"]]}');
4    //res can be leaked to malicious server
5    ... ...
6    },1500);
7    window.location.href = "https://privileged.
     com/";
```

Figure: Example for verifying domain name confusion.

Prevalence & Consequence

- Step II: Vulnerability Analysis
  - Confusion Overall Result
    - *all* vulnerable to at least one type of identity confusion attack

Table 5: Breakdown of Identity Confusion Vulnerabilities of 47 Super-apps

Identity Confusion		# Super-apps	Examples
Domain	Type 1: Timing-based	15	WeChat, Alipay
	Type 2: Frame-based	15	Microsoft Teams, Go-Jek
	Total	15	
AppID	Type 1: Flawed matching	26	TikTok, Baidu
	Type 2: Flawed parsing	2	WeChat, Go-Jek
	Type 3: Missing checks	10	Microsoft Teams, UnionPay
	Total	38	
Capability	Type 1: Client-side	1	UnionPay
	Type 2: Server-side	1	WeChat
	Total	2	
No identity checks		9	Snapchat, Kuaishou
Total		47	

Prevalence & Consequence

- Step III: Consequence Analysis
  - privilege escalation
  - phishing
  - privacy leaks

Table 8: Breakdown of Identity Confusion Consequences of 47 Super-apps

Consequences	# Super-apps	Examples
Privilege Escalation	38	Go-Jek, Grab
Phishing	31	TikTok, WeChat
Privacy Leaks	35	Alipay, Microsoft Teams

More in the Paper

- Two other identity confusions
  - App ID Confusion
  - Capability Confusion
- Overall Result Details
  - flaws & consequences in total 47 super-apps
  - other three consequences
    - permission re-delegation
    - data leakage
    - data over-collection
- Real-world Case Studies
- Mitigation & Discussion

Conclusion

- Conduct the first systematic study on identity confusion vulnerabilities in WebView-based app-in-app ecosystem.
- We collect and analyze 47 popular real-world super-apps, and confirms that they are all vulnerable to different types of identity confusion vulnerabilities
- We thoroughly study why such identity confusion vulnerabilities exist and propose corresponding mitigation strategies based on the causes

Thanks !

Q&A

zhibozhang19@fudan.edu.cn