



Unpacking the Threats of All-in-One Mobile Super Apps

Zhiqiang Lin

zlin@cse.ohio-state.edu

October 17th, 2023

Outline

1 Introduction

2 Threats from Vulnerability Exploitation

3 Threats from Malware Attacks

4 Takeaway

Outline

1 Introduction

2 Threats from Vulnerability Exploitation

3 Threats from Malware Attacks

4 Takeaway

Acknowledgement



Yue Zhang



Bayan Turkistani



Chaoshun Zuo



Yuqing Yang



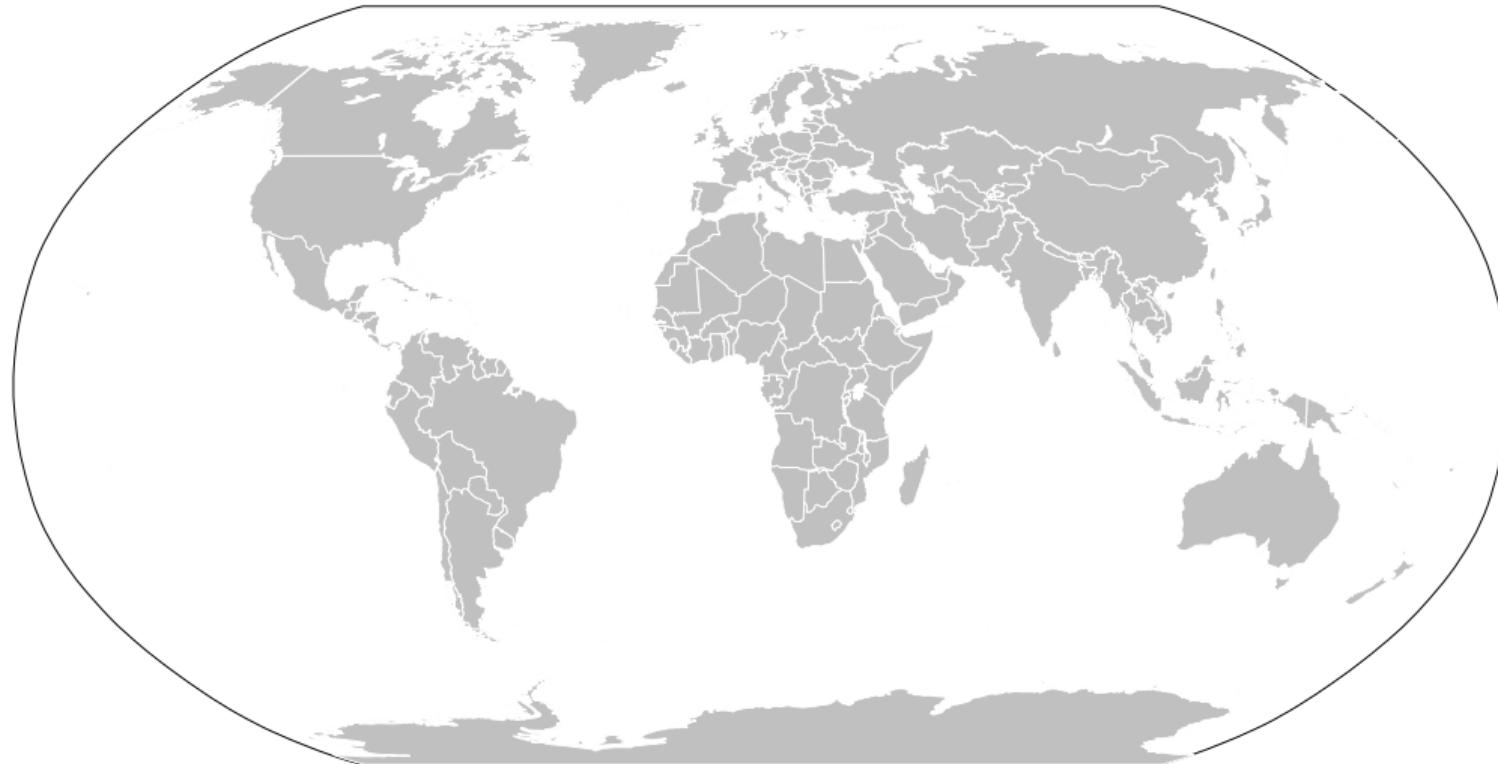
Chao Wang



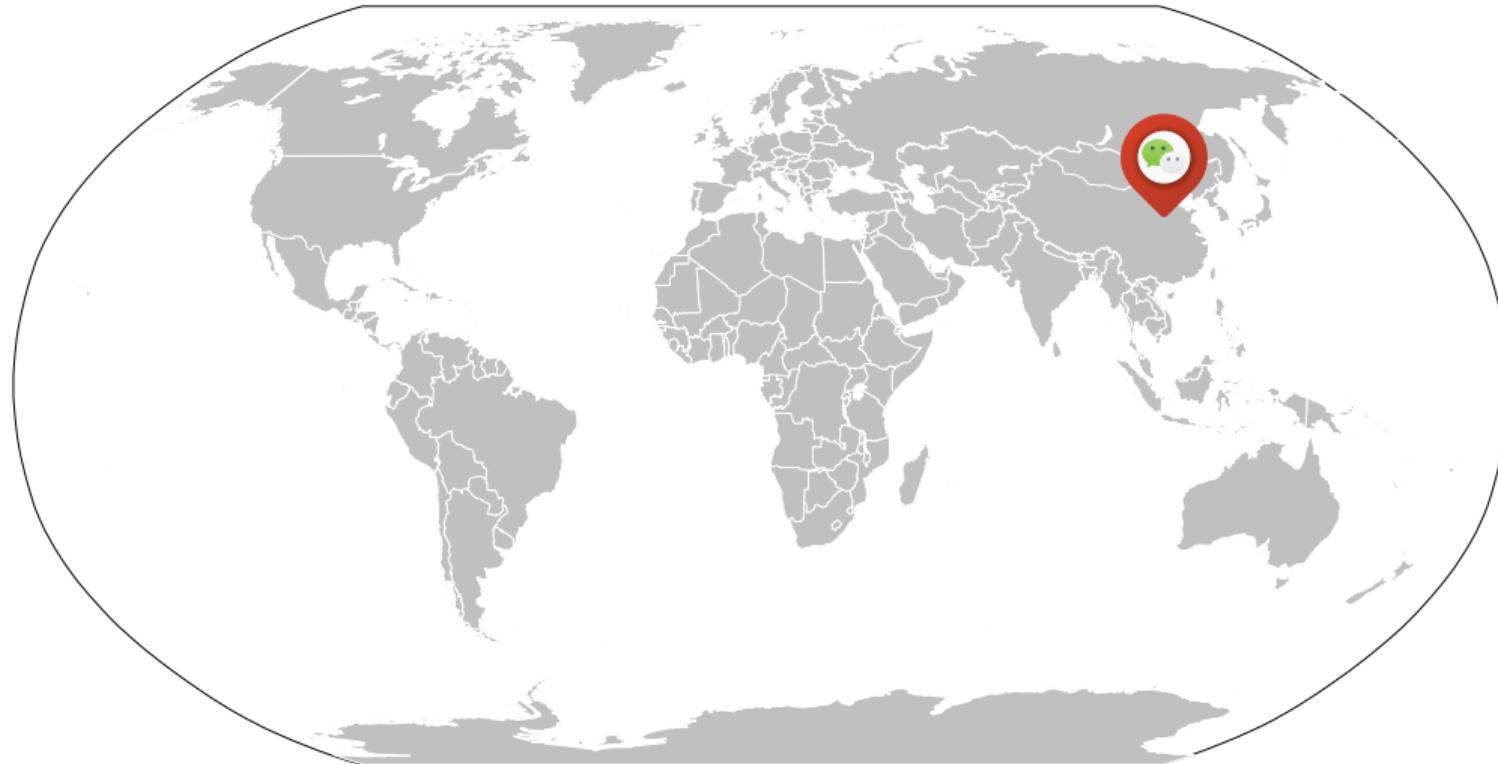
Banana Kn.

- ① Uncovering and Exploiting Hidden APIs in Mobile Super Apps. In CCS 2023 [WZL23b]
 - ② Don't Leak Your Keys: Understanding, Measuring, and Exploiting the AppSecret Leaks in Mini-Programs. In CCS 2023 [ZYL23]
 - ③ One Size Does Not Fit All: Uncovering And Exploiting Cross Platform Discrepant APIs in Wechat. In USENIX Security 2023 [WZL23a]
 - ④ TAINTMINI: Detecting Flow of Sensitive Data in Mini-Programs with Static Taint Analysis. In ICSE 2023 [WKZ⁺]
 - ⑤ Cross Miniapp Request Forgery: Root Causes, Attacks, and Vulnerability Detection. In CCS 2022 [YZL22]
 - ⑥ A Measurement Study of Wechat Mini-Apps. In SIGMETRICS 2021 [ZTY⁺21]

The World of Mobile Super Apps (“One App with Multiple Services”)



The World of Mobile Super Apps (“One App with Multiple Services”)



The World of Mobile Super Apps (“One App with Multiple Services”)



The World of Mobile Super Apps (“One App with Multiple Services”)



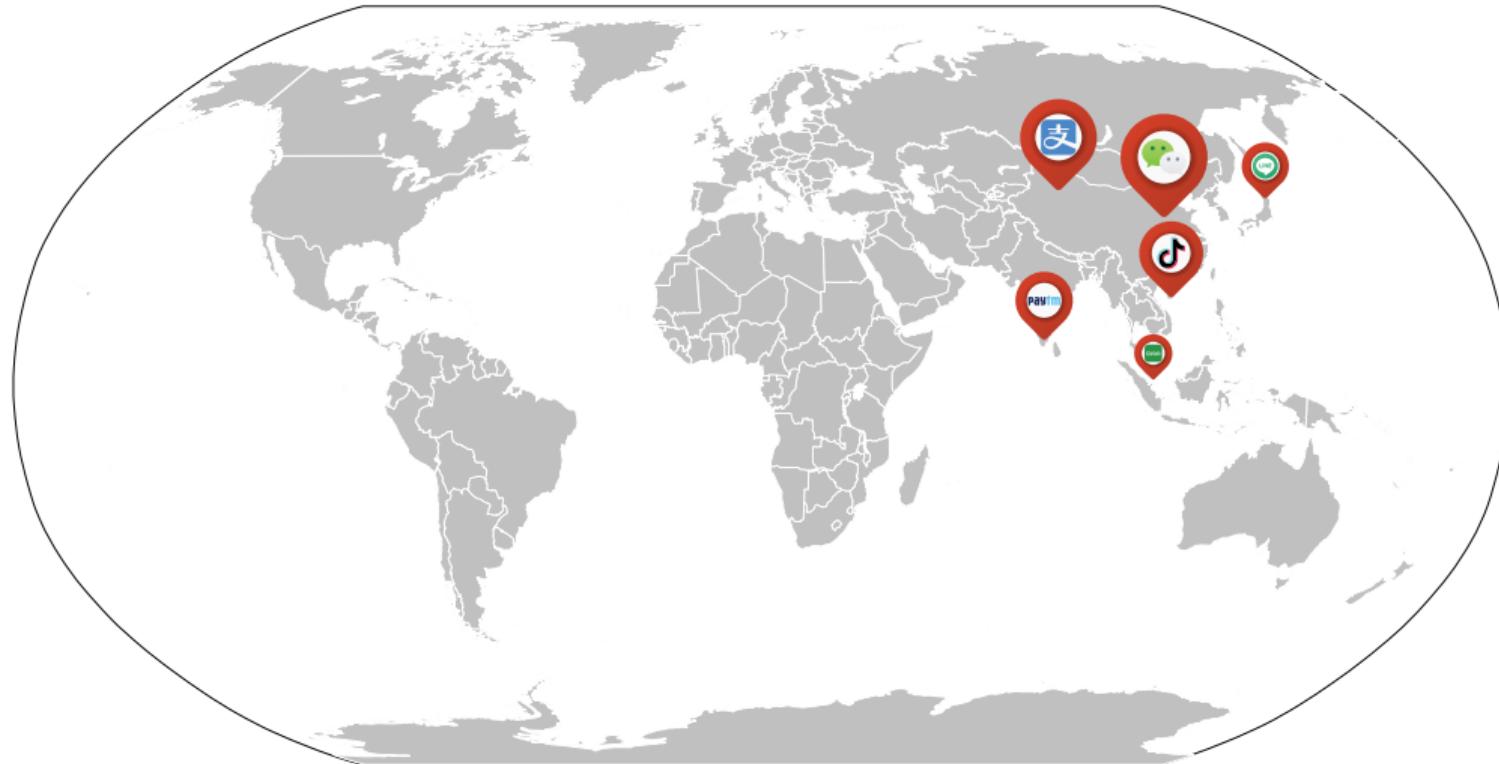
The World of Mobile Super Apps (“One App with Multiple Services”)



The World of Mobile Super Apps (“One App with Multiple Services”)



The World of Mobile Super Apps (“One App with Multiple Services”)



The World of Mobile Super Apps (“One App with Multiple Services”)



The World of Mobile Super Apps (“One App with Multiple Services”)



The World of Mobile Super Apps (“One App with Multiple Services”)



The World of Mobile Super Apps (“One App with Multiple Services”)

Super App	Category	Monthly Users	Country	Services										Platform	Miniapp	
				Business	Education	Communication	Finance	Food Delivery	Games	Lifestyle	Ride-hailing	Shopping	Social			
WeChat	Social	1,200 million +	China	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Tiktok	Social	1,000 million +	China	✗	✓	✓	✓	✗	✓	✓	✗	✓	✓	✓	✓	✓
Alipay	Finance	730 million +	China	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Snapchat	Social	347 million +	U.S.	✗	✗	✓	✗	✗	✓	✓	✗	✗	✓	✓	✗	✓
WeCom	Business	180 million +	China	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Paytm	Finance	150 million +	India	✓	✗	✗	✓	✗	✗	✗	✗	✗	✓	✓	✓	✗
Go-Jek	Finance	100 million +	Indonesia	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
Zalo	Social	52 million +	Vietnam	✓	✗	✓	✗	✗	✓	✗	✗	✓	✓	✓	✓	✗
Kakao	Social	45 million +	South Korea	✗	✗	✓	✗	✓	✗	✓	✓	✓	✓	✓	✓	✗
Grab	Delivery	25 million +	Singapore	✗	✗	✗	✓	✓	✗	✓	✓	✓	✗	✓	✓	✗

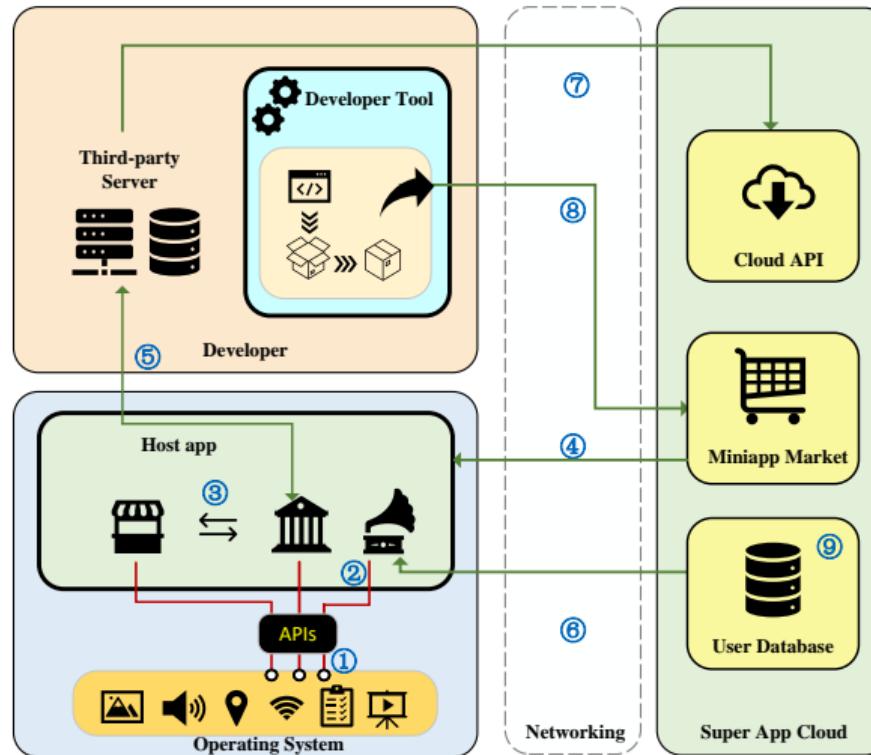
What is WeChat?

"It's sort of like Twitter, plus PayPal, plus a whole bunch of things all rolled into one, with a great interface."

— Elon Musk



Mobile Superapps in a Nutshell

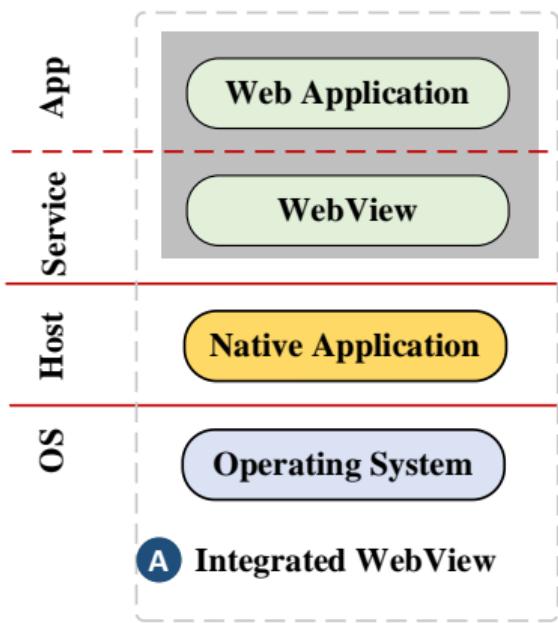


The Benefits a Superapp Can Offer

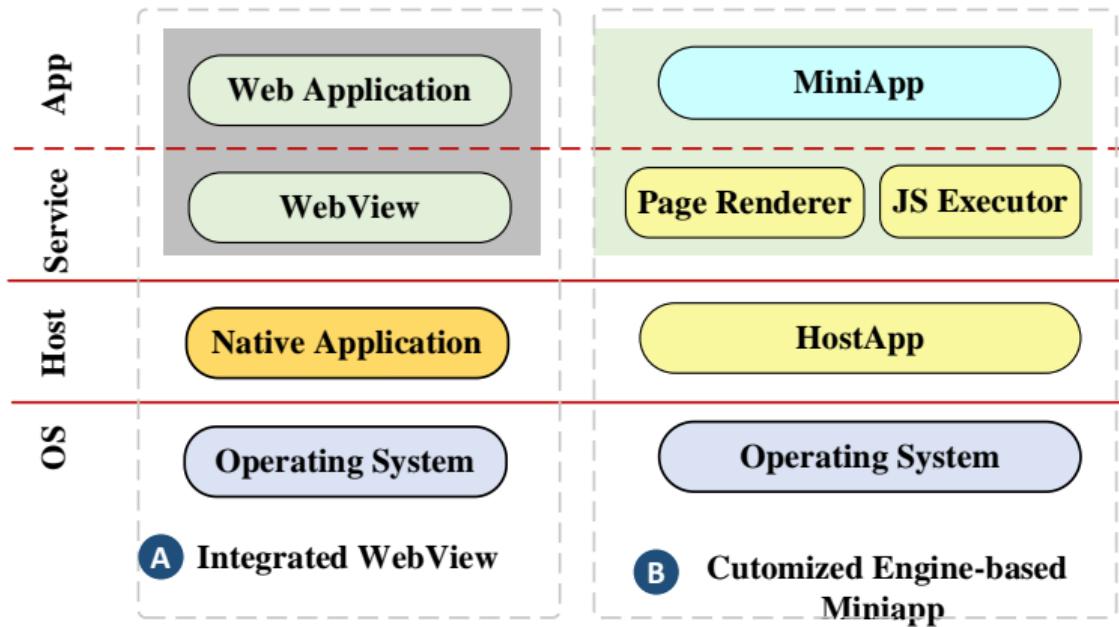
Hosts	Mobile OS (Native Apps)	Web Browsers (Web Apps)	Super Apps (Miniapps)
Example Platform	Android	Chrome	WeChat
System Resources?	●	○	●
Super-app Services?	○	●	●
User Data/States?	●	●	●
Account?	●	●	●
App Packages?	●	○	●
Cloud Services?	○	●	●
API Support?	Rich	Poor	Rich
Compatible with Platforms?	○	●	●
Backend?	●	●	○
Centralized Vetting?	●	○	●
Install-free?	○	●	●
Market?	●	○	●
Storage Consumption?	High	Low	Low
Update?	Client-based	Client-based	Server-based
Performance?	High	Browser-specific	Super-app-specific
Offline Loading?	High	Low	Median
Register and Login?	●	●	○

"●" represents full support; "○" represents partial support; "○" represents no support.

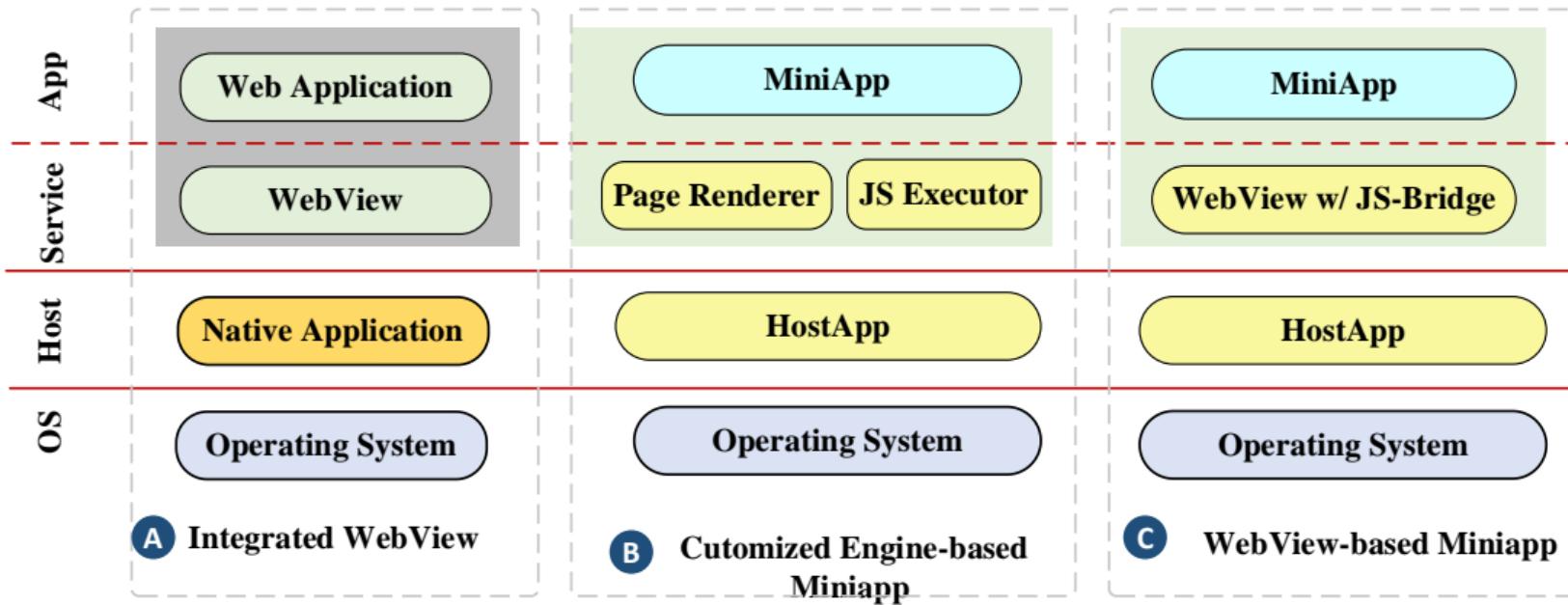
The Taxonomy of Super Apps



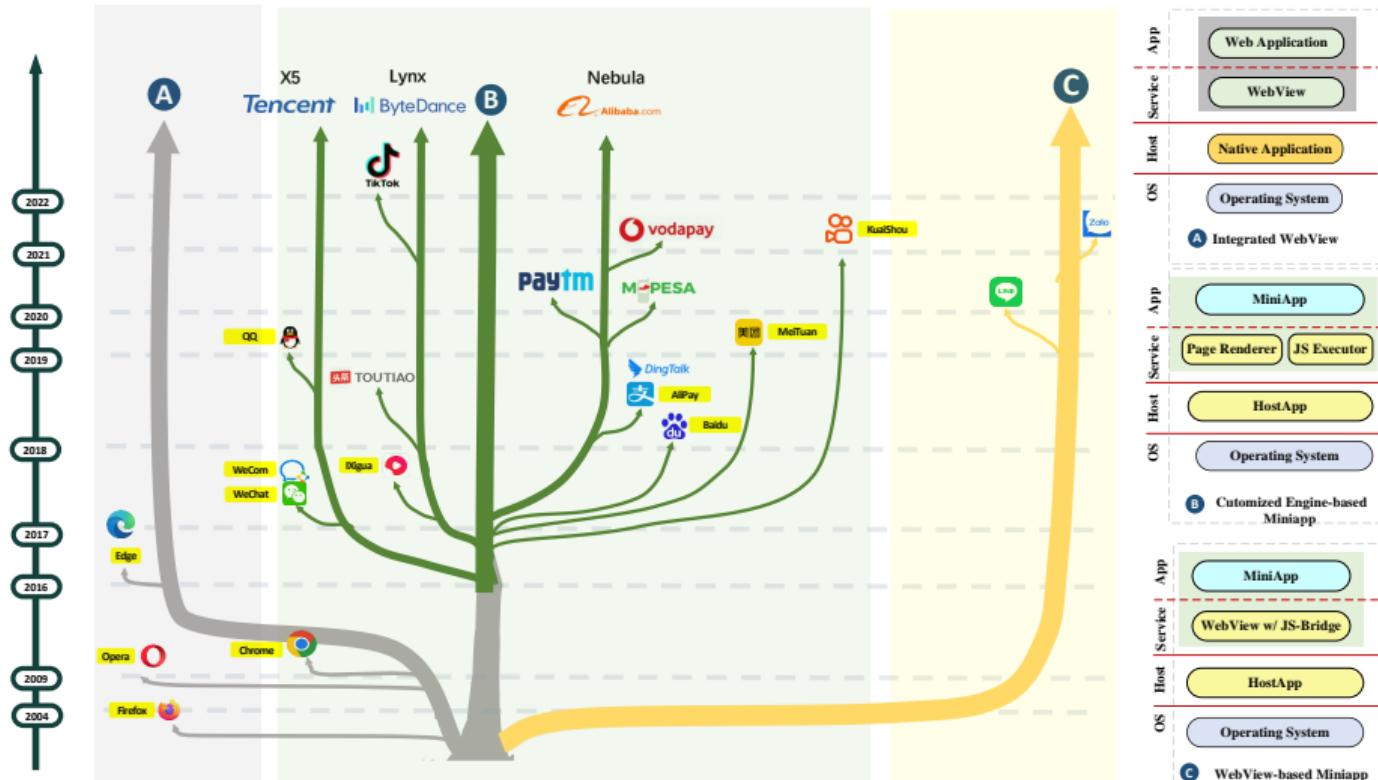
The Taxonomy of Super Apps



The Taxonomy of Super Apps



Evolution of the Superapps



Security Threats

Threats from Vulnerability Exploitation

① Vulnerabilities in Host Apps

- (T1) Platform Discrepancies [WZL23a]
- (T2) Privileged APIs [WZL23b]
- (T3) Identity Confusion [ZZL⁺²²]

② Vulnerabilities in Miniapps

- (T4) Cross Miniapp Request Forgery [YZL22]
- (T5) AppSecret Key Leakage [ZYL23]
- (T6) Missing Signature Verification [ZZW23]

Threats from Malware Attacks

① API Misuse/Abuse (Payload)

- (T7) Collecting User Privacy
- (T8) Service Abusing
- (T9) Grayware

② Bypassing Vetting

- (T10) Code Vetting Bypassing
- (T11) Content Vetting Bypassing
- (T12) Reporting Bypassing

Outline

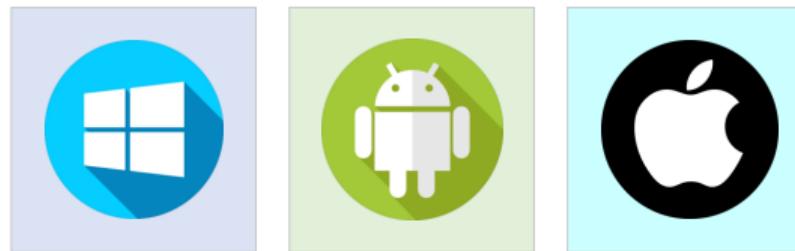
1 Introduction

2 Threats from Vulnerability Exploitation

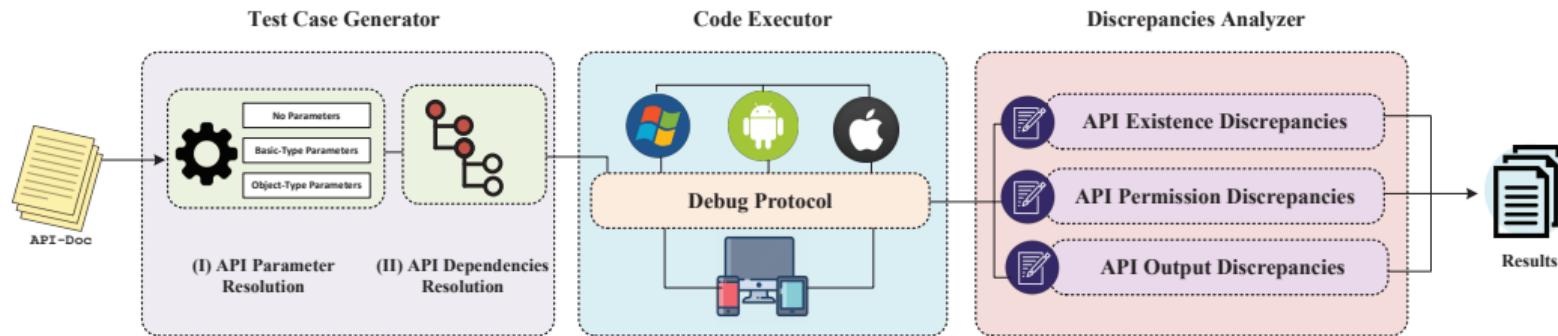
3 Threats from Malware Attacks

4 Takeaway

(T1) Exploiting Cross-platform Discrepancies [WZL23a]



(T1) Exploiting Cross-platform Discrepancies [WZL23a]



(T1) Exploiting Cross-platform Discrepancies [WZL23a]

APIs	Permission Scope	Mobile				PC	
		Android		iOS		Windows	
		A	P	A	P	A	P
getLocation		✓	✓	✓	✓	✓	✗
chooseLocation	userLocation	✓	✓	✓	✓	✓	✗
startLocationUpdate		✓	✓	✓	✓	✓	✗
SLUBackground*	userLocationBackground	✓	✓	✓	✓	✗	-
startRecord		✓	✓	✓	✓	✓	✗
joinVoIPChat	record	✓	✓	✓	✓	✗	-
RecorderManager.start		✓	✓	✓	✓	✓	✗
createCameraContext		✓	✓	✓	✓	✓	✗
createVKSession	camera	✓	✓	✓	✓	✗	-
openBluetoothAdapter		✗	-	✓	✓	✗	-
BLEPeripheralServer	bluetooth	✓	✓	✓	✓	✗	-
saveImageToPhotosAlbum		✓	✓	✓	✓	✓	✗
saveVideoToPhotosAlbum	writePhotosAlbum	✓	✓	✓	✓	✓	✗
addPhoneContact	addPhoneContact	✓	✓	✓	✓	✗	-
addPhoneRepeatCalendar		✓	✓	✓	✓	✗	-
addPhoneCalendar	addPhoneCalendar	✓	✓	✓	✓	✗	-
getWeRunData	werun	✓	✓	✓	✓	✗	-

(T1) Exploiting Cross-platform Discrepancies [WZL23a]

Name	Category	Type	Precision	Mobile			Desktop			
				A	S	U	A	S	U	
createAudioContext	Media			✓		✓	✓	✓		✓
createBufferURL	Storage			✓		✓	✓		✓	✓
createCameraContext	Media			✓		✓	✓		✓	
createCanvasContext	Canvas			✓		✓	✓		✓	
createIntersectionObserver	WXML			✓		✓	✓		✓	
createLivePusherContext	Media			✓		✓	✓		✓	
createOffscreenCanvas	Canvas			✓		✓	✓		✓	
createSelectorQuery	WXML			✓		✓	✓		✓	
createWebAudioContext	Media			✓		✓	✓		✓	
getAccountInfoSync	OpenAPI			✓	✓		✓	✓	✓	
getAppAuthorizeSetting	Base			✓	✓	✓	✓	✓	✓	
getAppBaseInfo	Base			✓	✓	✓	✓	✓	✓	
getDeviceInfo	Base			✓	✓	✓	✓	✓	✓	
getLocalIPAddress	Device			✓	✓	✓	✓	✓		
getMenuItemBoundingClientRect	UI			✓	✓		✓	✓	✓	
getPerformance	Base			✓	✓	✓	✓	✓		
getScreenBrightness	Device			✓	✓	✓	✓	✓		
getSystemInfo	Base			✓	✓	✓	✓	✓	✓	
getSystemInfoAsync	Base			✓	✓	✓	✓	✓	✓	
getSystemInfoSync	Base			✓	✓	✓	✓	✓	✓	
getSystemSetting	Base			✓	✓	✓		✓	✓	
getWindowInfo	Base			✓	✓	✓		✓	✓	

(T2) Exploiting Hidden/Privileged APIs [WZL23b]

Attacks Caused by Hidden APIs

- ① Arbitrary Web Page Access
- ② Malware Download and Installation
- ③ Screenshot-based Information Theft
- ④ Phone Number Theft
- ⑤ Contact Information Theft

```
1 // Documented API Implementation of Baidu
2 package com.baidu.swan.apps.scheme.actions.f;
3 public class a extends aa {
4     public a (e context) {
5         super(context, "/swanAPI/getLocation");
6     }
7
8     @Override
9     public boolean a (Context c, Scheme s, CallbackHandler cb, SwanApp a) {
10        // some other logic
11    }
12 }
13
14 // Unocumented API Implementation of Baidu
15 package com.baidu.swan.apps.impl.account.a;
16 public class f extends aa {
17     public f (e context) {
18         super(context, "/swanAPI/getBDUSS");
19     }
20
21     @Override
22     public boolean a (Context c, Scheme s, CallbackHandler cb, SwanApp a) {
23        // some other logic
24    }
25 }
```

(T2) Exploiting Hidden/Privileged APIs [WZL23b]

The diagram illustrates the flow of exploit code across three layers:

- JavaScript Framework Layer:** Contains the `WeixinJSBridge` function.
- Service Abstraction Layer:** Contains the `MBRuntime` class.
- Customized V8 Layer:** Contains the `magicbrush` implementation.

The exploit flow starts in the `WeixinJSBridge` function, which calls `NativeGlobal.invokeHandler`. This leads to the `MBRuntime.nativeInvokeHandler` method in the Service Abstraction Layer. Finally, it reaches the `magicbrush::nativeglobal::invokeHandler` function in the Customized V8 Layer, which then calls `magicbrush::nativeglobal::invokeHandler(v8::Isolate *a1, DWORD *a2)`.

```

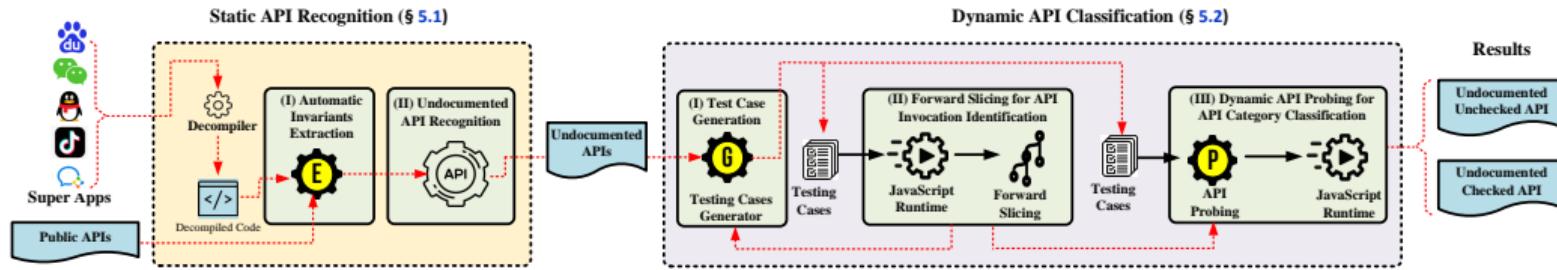
    graph TD
        JSF[JavaScript Framework Layer] --> SAI[Service Abstraction Layer]
        SAI --> CVL[Customized V8 Layer]
    
```

```

    // JavaScript Framework Layer
    1 WeixinJSBridge = function(global) {
    2     var NativeGlobal = global.NativeGlobal;
    3     var globalCount = 0;
    4
    5     function invokeMethod(apiName, params, callbackHandler) {
    6         params = WeixinNativeBuffer.pack(params);
    7         var filteredParams = paramFilter(params || {});
    8         callbackId = ++globalCount;
    9         callbackQueue[callbackId] = callbackHandler,
    10            a(apiName, params, callbackId) {
    11                callbackId = NativeGlobal.invokeHandler(apiName, params,
    12                    callbackId);
    13                invokeCallbackHandler(callbackId, callbackHandler)
    14            }(apiName, filteredParams, callbackId)
    15        }
    16        return this;
    17 }(global);

    // Implementation of invoke handler in Java framework
    1 // Implementation of invoke handler in NativeGlobal JavaScript Object (C++)
    2 int magicbrush::BindingNativeGlobal::BindTo<v8::Object *a1, v8::Isolate *a2>{
    3     /* Code Omitted */
    4
    5     v13 = 0;
    6     v7 = (v8::Value *)mm::JSGet<v8::Local<v8::Value>>(a1, v6, "NativeGlobal", &v12);
    7     if ( !v7 || (v9 = (int)v7, !v8::Value::IsObject(v7)) )
    8         v9 = v8::Object::New(a1, v8);
    9     v13 = v9;
    10
    11    /* Code Omitted */
    12
    13    mm::JSSet<v8::Local<v8::Object>>(a1, *a3, "NativeGlobal", v13);
    14    return v13;
    15
    16    int magicbrush::nativeglobal::invokeHandler(v8::Isolate *a1, DWORD *a2) {
    17        /* Code Omitted */
    18
    19        mm::JSConvert<std::string, void>::fromV8(api_name, a1, v6);
    20        mm::JSConvert<char16_t const*, void>::fromV8(api_param, a1, v6);
    21        mm::JSConvert<int, void>::fromV8(callback_id, a1, v6);
    22        Java_com_tencent_magicbrush_MBRuntime_nativeInvokeHandler(
    23            api_name,
    24            api_param,
    25            callback_id
    26        )
    27
    28        /* Code Omitted */
    29
    30    }
    31
    32    /* Code Omitted */
    33
    34    /* Code Omitted */
    35
    }
```

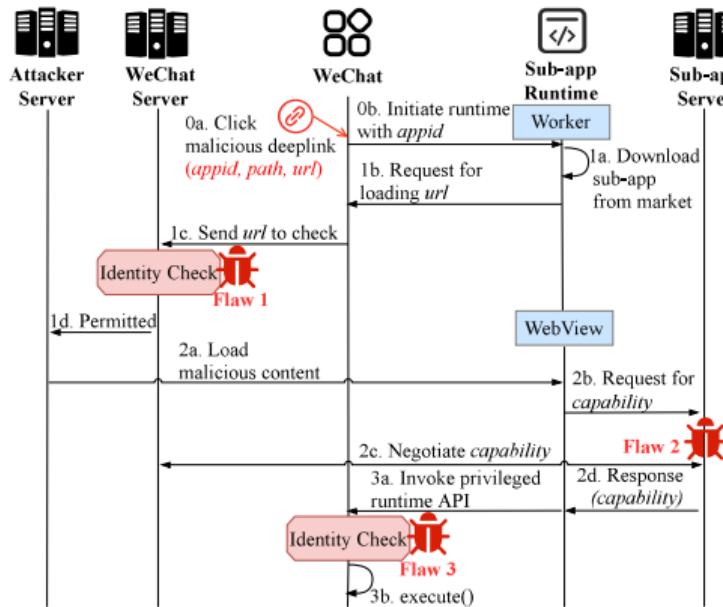
(T2) Exploiting Hidden/Privileged APIs [WZL23b]



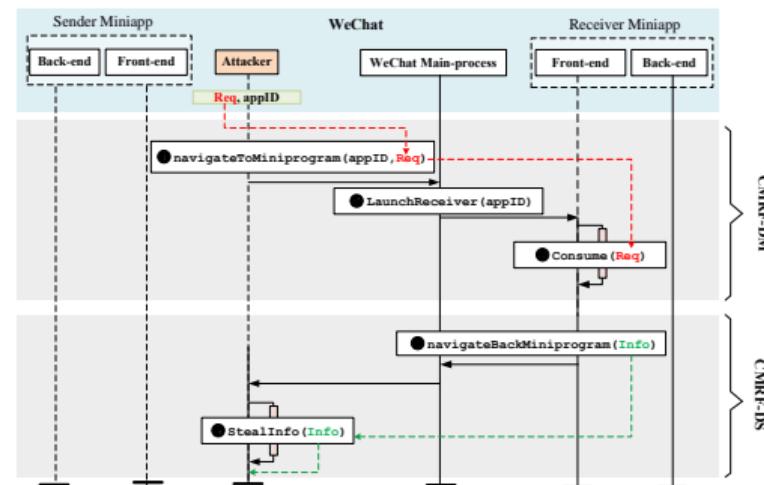
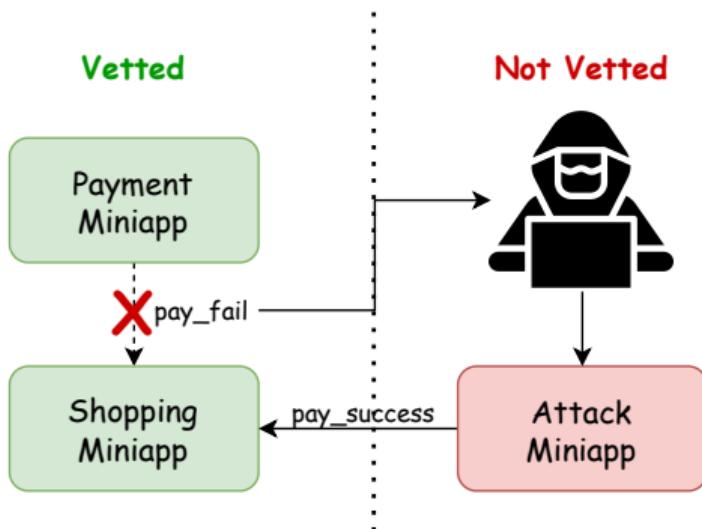
(T2) Exploiting Hidden/Privileged APIs [WZL23b]

Available APIs	WeChat				WeCom				Baidu				TikTok				QQ																
	D	%	UU	%	UC	%	D	%	UU	%	UC	%	D	%	UU	%	UC	%	D	%	UU	%	UC	%									
Base	Basic	5	71.4	2	28.6	-	0.0	6	66.7	3	33.3	-	0.0	8	72.7	2	18.2	1	9.1	7	63.6	4	36.4	-	0.0	3	100.0	-	0.0	-	0.0		
	App	13	39.4	14	42.4	6	18.2	13	37.1	16	45.7	6	17.1	8	42.1	10	52.6	1	5.3	6	50.0	6	50.0	-	0.0	9	34.6	17	65.4	-	0.0		
	Debug	15	88.2	2	11.8	-	0.0	15	88.2	2	11.8	-	0.0	1	3.3	28	93.3	1	3.3	-	0.0	-	0.0	-	0.0	20	100.0	-	0.0	-	0.0		
	Misc	10	58.8	7	41.2	-	0.0	10	55.6	8	44.4	-	0.0	9	100.0	-	0.0	0.0	10	52.6	9	47.4	-	0.0	9	100.0	-	0.0	-	0.0			
UI	Interaction	6	46.2	7	53.8	-	0.0	6	46.2	7	53.8	-	0.0	7	41.2	10	58.8	-	0.0	9	81.8	2	18.2	-	0.0	6	40.0	9	60.0	-	0.0		
	Navigation	4	44.4	5	55.6	-	0.0	4	40.0	6	60.0	-	0.0	4	100.0	-	0.0	-	0.0	5	100.0	-	0.0	-	0.0	4	33.3	8	66.7	-	0.0		
	Animation	32	100.0	-	0.0	0.0	32	100.0	-	0.0	-	0.0	0.0	21	95.5	1	4.5	-	0.0	1	100.0	-	0.0	-	0.0	31	100.0	-	0.0	-	0.0		
	WebView	-	0.0	22	95.7	1	4.3	-	0.0	24	96.0	1	4.0	-	0.0	3	75.0	1	25.0	-	0.0	3	100.0	-	0.0	-	0.0	16	100.0	-	0.0	-	0.0
	Misc	20	27.0	54	73.0	-	0.0	20	25.6	58	74.4	-	0.0	37	77.1	11	22.9	-	0.0	14	73.7	5	26.3	-	0.0	18	42.0	24	57.1	-	0.0		
Network	Request	5	55.6	4	44.4	-	0.0	5	55.6	4	44.4	-	0.0	2	66.7	1	33.3	-	0.0	6	60.0	4	40.0	-	0.0	4	66.7	2	33.3	-	0.0		
	Download	7	24.1	21	72.4	1	3.4	7	23.3	22	73.3	1	3.3	11	100.0	-	0.0	-	0.0	0	0.0	4	100.0	-	0.0	6	60.0	4	40.0	-	0.0		
	Upload	7	50.0	5	35.7	2	14.3	7	46.7	6	40.0	2	13.3	6	100.0	-	0.0	-	0.0	0	0.0	4	100.0	-	0.0	6	75.0	2	25.0	-	0.0		
	Websocket	14	93.3	1	6.7	-	0.0	14	93.3	1	6.7	-	0.0	13	100.0	-	0.0	-	0.0	7	77.8	2	22.2	-	0.0	13	86.7	2	13.3	-	0.0		
Storage	Misc	23	88.5	3	11.5	-	0.0	23	85.2	4	14.8	-	0.0	0.0	0.0	-	0.0	-	0.0	0.0	-	0.0	0.0	10	55.6	8	44.4	-	0.0				
	Map	10	66.7	5	33.3	-	0.0	10	66.7	5	33.3	-	0.0	10	100.0	-	0.0	-	0.0	10	90.0	1	9.1	-	0.0	10	83.3	2	16.7	-	0.0		
	Image	8	14.3	48	85.7	-	0.0	8	14.3	48	85.7	-	0.0	7	100.0	-	0.0	-	0.0	6	100.0	-	0.0	-	0.0	9	36.0	16	64.0	-	0.0		
	Video	6	60.0	4	40.0	-	0.0	6	60.0	4	40.0	-	0.0	6	85.7	1	14.3	-	0.0	5	83.3	1	16.7	-	0.0	6	60.0	4	40.0	-	0.0		
Media	Audio	14	35.0	26	65.0	-	0.0	14	31.8	30	68.2	-	0.0	19	95.0	1	5.0	-	0.0	8	80.0	2	20.0	-	0.0	14	63.6	8	36.4	-	0.0		
	Live	64	84.2	9	11.8	3	3.9	64	79.0	14	17.3	3	3.7	44	100.0	-	0.0	-	0.0	44	81.5	10	18.5	-	0.0	61	85.9	10	14.1	-	0.0		
	Recorder	26	46.4	30	53.6	-	0.0	26	39.4	40	60.6	-	0.0	8	100.0	-	0.0	-	0.0	19	100.0	-	0.0	-	0.0	23	57.5	17	42.5	-	0.0		
	Camera	16	84.2	3	15.8	-	0.0	16	84.2	3	15.8	-	0.0	12	100.0	-	0.0	-	0.0	11	91.7	1	8.3	-	0.0	15	88.2	2	11.8	-	0.0		
Open API	Misc	9	60.0	6	40.0	-	0.0	9	52.9	8	47.1	-	0.0	9	50.0	9	50.0	-	0.0	20	95.2	1	4.8	-	0.0	4	36.4	7	63.6	-	0.0		
	Location	3	42.9	4	57.1	-	0.0	3	42.9	4	57.1	-	0.0	7	100.0	-	0.0	-	0.0	3	100.0	-	0.0	-	0.0	3	100.0	-	0.0	-	0.0		
	Share	4	33.3	7	58.3	1	8.3	4	16.7	19	79.2	1	4.2	3	100.0	-	0.0	-	0.0	5	71.4	2	28.6	-	0.0	5	35.7	9	64.3	-	0.0		
	Canvas	60	74.1	21	25.9	-	0.0	60	74.1	21	25.9	-	0.0	46	92.0	4	8.0	-	0.0	49	98.0	1	2.0	-	0.0	48	92.3	4	7.7	-	0.0		
File	File	39	97.5	1	2.5	-	0.0	39	92.9	3	7.1	-	0.0	39	100.0	-	0.0	-	0.0	34	97.1	1	2.9	-	0.0	37	97.4	1	2.6	-	0.0		
	Login	2	100.0	-	0.0	0.0	5	83.3	1	16.7	-	0.0	3	42.9	1	14.3	3	42.9	2	100.0	-	0.0	-	0.0	2	100.0	-	0.0	-	0.0			
	Navigate	2	33.3	2	33.3	2	33.3	2	22.2	5	55.6	2	22.2	3	100.0	-	0.0	-	0.0	7	100.0	-	0.0	-	0.0	2	50.0	1	25.0	1	25.0		
	User Info	2	16.7	7	58.3	3	25.0	5	23.8	13	61.9	3	14.3	1	10.0	6	60.0	3	30.0	2	13.3	13	66.7	-	0.0	2	28.6	4	57.1	1	14.3		
Open API	Payment	1	3.4	13	44.8	15	51.7	1	3.2	15	48.4	15	48.4	1	50.0	-	0.0	1	50.0	1	33.3	1	33.3	2	22.2	7	77.8	-	0.0	-	0.0		
	Bio-Auto	3	27.3	3	27.3	5	45.5	3	21.4	6	42.9	5	35.7	-	0.0	-	0.0	-	0.0	1	100.0	-	0.0	-	0.0	3	100.0	-	0.0	-	0.0		
	Enterprise	-	0.0	1	100.0	-	0.0	5	17.9	6	21.4	17	60.7	-	0.0	-	0.0	-	0.0	0	0.0	-	0.0	-	0.0	0	0.0	-	0.0	-	0.0		
	Misc	14	19.4	42	58.3	16	22.2	14	16.7	54	64.3	16	19.0	16	16.7	51	7.2	1	7.1	10	35.7	25	55.6	20	44.4	-	0.0	12	13.0	78	84.8	2	2.2
Device	Wi-Fi	9	100.0	-	0.0	0	0	9	100.0	-	0.0	-	0.0	10	100.0	-	0.0	-	0.0	4	100.0	-	0.0	-	0.0	9	100.0	-	0.0	-	0.0		
	Bluetooth	18	60.0	11	36.7	1	3.3	18	58.1	12	38.7	1	3.2	-	0.0	-	0.0	-	0.0	0	0.0	-	0.0	-	0.0	18	100.0	-	0.0	-	0.0		
	Contact	1	10.0	5	50.0	4	40.0	1	9.1	6	54.5	4	36.4	1	33.3	2	66.7	-	0.0	-	0.0	-	0.0	-	0.0	1	25.0	2	50.0	1	25.0		
	NFC	5	26.3	14	73.7	-	0.0	9	39.1	14	60.9	-	0.0	0	0	-	0.0	-	0.0	0	0.0	-	0.0	-	0.0	5	100.0	-	0.0	-	0.0		
AI	Screen	4	36.4	6	54.5	1	9.1	4	36.4	6	54.5	1	9.1	3	100.0	-	0.0	-	0.0	9	100.0	-	0.0	-	0.0	4	100.0	-	0.0	-	0.0		
	Phone	1	4.3	21	91.3	1	3.1	4	21	91.3	1	4.3	1	100.0	-	0.0	-	0.0	0	1	100.0	-	0.0	-	0.0	1	50.0	1	50.0	-	0.0		
	Misc	28	63.6	15	34.1	1	2.3	28	59.6	18	38.3	1	2.1	21	80.8	5	19.2	-	0.0	16	69.6	7	30.4	-	0.0	28	82.4	6	17.6	-	0.0		
	CV	19	100.0	-	0.0	0	0	19	100.0	-	0.0	-	0.0	18	90.0	2	10.0	-	0.0	-	0.0	-	0.0	-	0.0	0	0.0	-	0.0	-	0.0		
AD	19	95.0	1	5.0	-	0.0	19	95.0	1	5.0	-	0.0	9	64.3	4	28.6	1	7.1	13	61.9	8	38.1	-	0.0	3	25.0	9	75.0	-	0.0			
	Uncategorized	30	38.5	47	60.3	1	30	38.6	51	60.3	15	23.8	1	15.6	13	53.6	10	35.7	3	10.7	17	68.0	7	28.0	1	4.0	34	68.0	15	30.0	1	2.0	
All	504	51.0	502	43.4	65	5.6	600	47.3	502	46.3	82	1.4	6.4	464	77.1	113	18.8	25	4.2	383	75.8	120	23.8	2	0.4	506	62.7	295	36.6	6	0.7		

(T3) Exploiting Identity Confusion Vulnerability [ZZL⁺22]



(T4) Cross Miniapp Request Forgery (CMRF) [YZL22]



(T4) Cross Miniapp Request Forgery (CMRF)

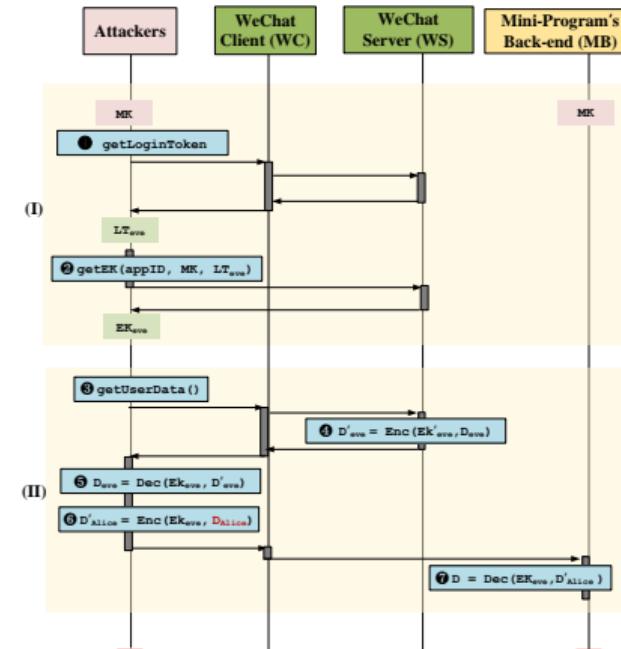
Category	WECHAT					
	No Use		Checked		Vulnerable	
	# app	% total	# app	%	# app	%
Business	131,078	5.1	81	8.07	923	91.93
E-learning	10,271	0.4	4	5.19	73	94.81
Education	240,077	9.34	184	3.72	4,756	96.28
Entertainment	29,442	1.14	140	33.02	284	66.98
Finance	3,509	0.14	6	6.67	84	93.33
Food	114,675	4.46	332	8.07	3,780	91.93
Games	88,056	3.42	10	2.09	469	97.91
Government	31,432	1.22	33	9.02	333	90.98
Health	27,716	1.08	37	5.44	643	94.56
Job	21,773	0.85	16	7.02	212	92.98
Lifestyle	394,493	15.34	269	4.23	6,092	95.77
Photo	9,039	0.35	3	4.41	65	95.59
Shopping	989,498	38.48	743	2.56	28,304	97.44
Social	20,671	0.8	6	2.99	195	97.01
Sports	15,980	0.62	69	22.48	238	77.52
Tool	261,467	10.17	122	3.72	3,161	96.28
Traffic	35,412	1.38	53	9.28	518	90.72
Travelling	10,524	0.41	5	3.62	133	96.38
Uncategorized	83,983	3.27	0	0.0	18	100.0
Total	2,519,096	97.96	2,113	4.03	50,281	95.97

Category	BAIDU					
	No Use		Checked		Vulnerable	
	# app	% total	# app	%	# app	%
Automobile	356	0.24	0	0.0	2	100.0
Business	5,201	3.5	0	0.0	113	100.0
Charity	2	0.0	0	0	0	0
E-commerce	96	0.06	0	0	0	0
Education	1,378	0.93	0	0.0	3	100.0
Efficiency	10,852	7.31	0	0.0	1	100.0
Entertainment	195	0.13	1	11.11	8	88.89
Finance	45	0.03	0	0.0	2	100.0
Food	123	0.08	0	0	0	0
Government	282	0.19	0	0.0	5	100.0
Health	2	0.0	0	0	0	0
Information	1,736	1.17	0	0.0	6	100.0
IT tech	113	0.08	0	0	0	0
Lifestyle	1,818	1.22	0	0	0	0
Medical	97	0.07	0	0	0	0
News	4	0.0	0	0	0	0
Post service	163	0.11	0	0	0	0
Real estate	1,510	1.02	0	0	0	0
Shopping	116,093	78.17	0	0.0	327	100.0
Social	205	0.14	0	0	0	0
Sports	145	0.1	0	0	0	0
Tool	46	0.03	0	0	0	0
Traffic	226	0.15	0	0.0	1	100.0
Travelling	1,473	0.99	0	0	0	0
Uncategorized	5,857	3.94	0	0.0	25	100.0
Total	148,018	99.67	1	0.2	493	99.8

(T5) Exploiting Key Leakage from Miniapps [ZYL23]

Attack Procedure

- ▶ (I) Obtaining Attacker's Encryption Key (EK)
 - ▶ Obtain leaked Master Key (MK)
 - ▶ Query for EK with the MK
- ▶ (II) Sensitive Data Retrieval and/or Manipulation
 - ▶ Capture encrypted data
 - ▶ Decrypt with MK
 - ▶ Data manipulation
 - ▶ Re-encrypt and send to back-end



(T5) Key Leakage from Miniapps [ZYL23]

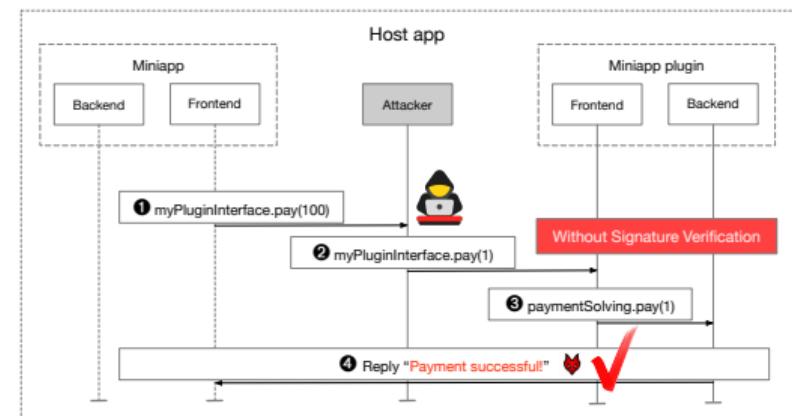
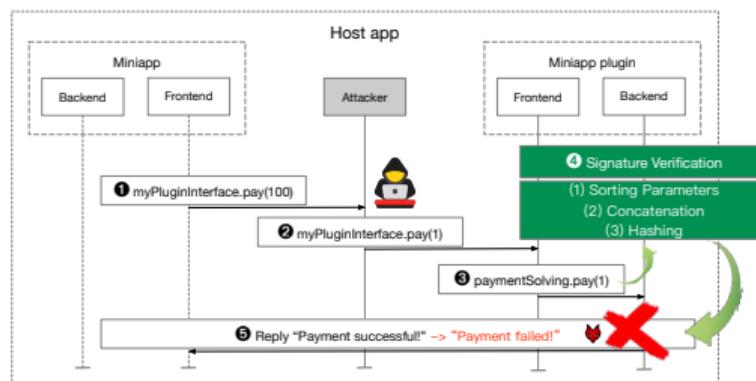
(I) Obtaining Attacker's Encryption Key (EK)



(II) User Phone Number Retrieval and Manipulation



(T6) Missing Signature Verification [ZYL23]



Tencent's Security Hall of Fame

Rank	Nickname	Link	Reputation	Credits
1	djurado		Experienced ...	113
2	Sergey Bobrov	https://twitter.com/Black2Fan	Proficient	46
3	OSU SecLab	https://seclab.engineering.osu.edu/	Proficient	46
4	kazan71p		Proficient	43
5	xCHCQg		Proficient	36
6	sh1yo	https://sh1yo.art	Proficient	32
7	NamHB		Proficient	31

<https://en.security.tencent.com/index.php/thanks>

Outline

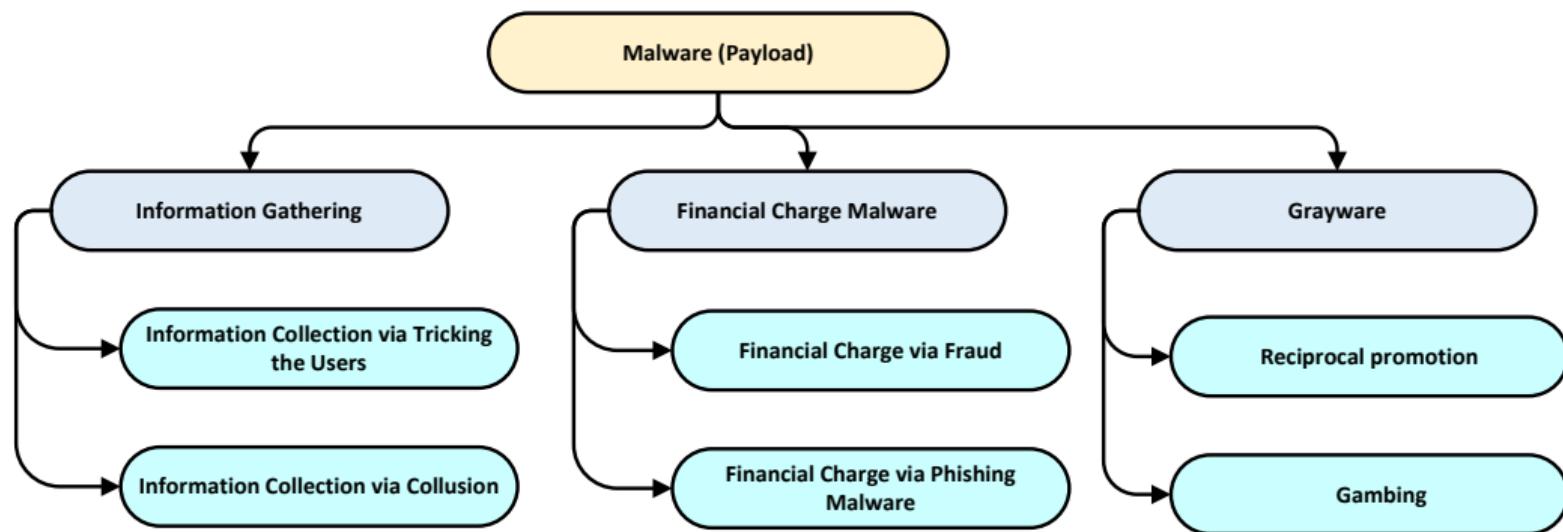
1 Introduction

2 Threats from Vulnerability Exploitation

3 Threats from Malware Attacks

4 Takeaway

Malware Taxonomy Based on Payloads



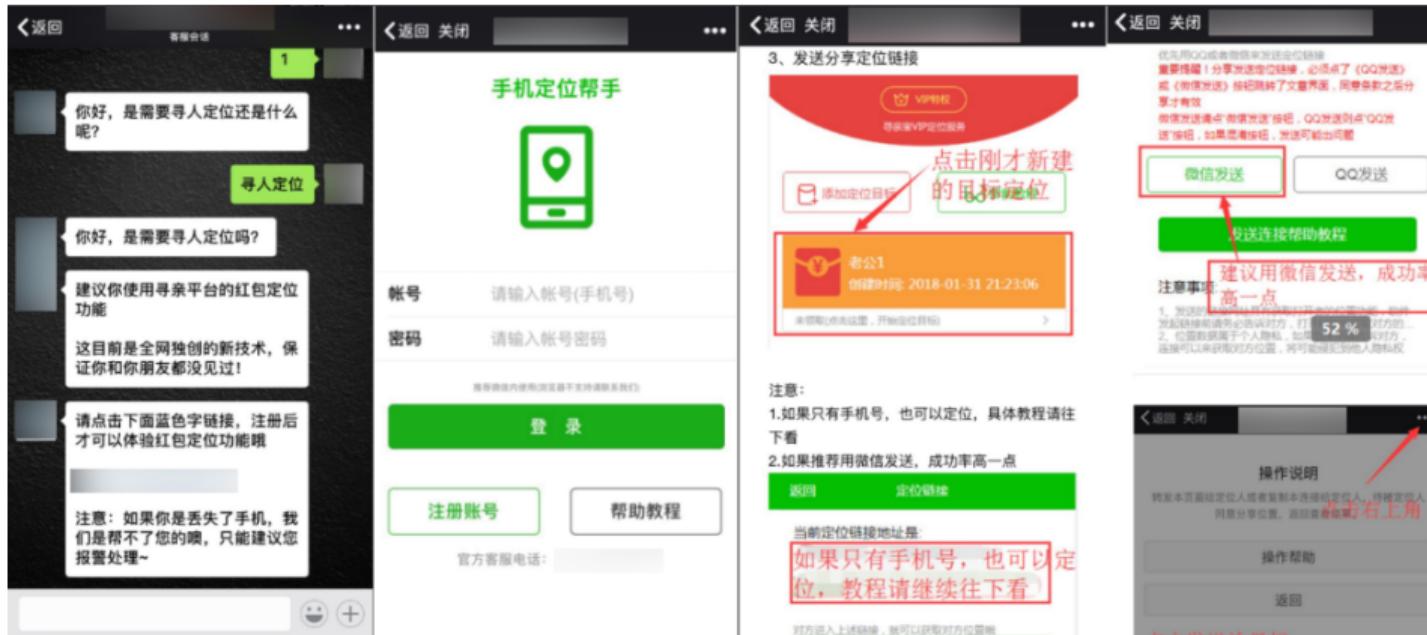
T7: Information Gathering

Information Gathering via Tricking the Users [opr]



T7: Information Gathering

Information Gathering via Collusion [opr]



T8: Financial Charge Malware

Financial Charge via Fraud [opr]

Fake Red Packet

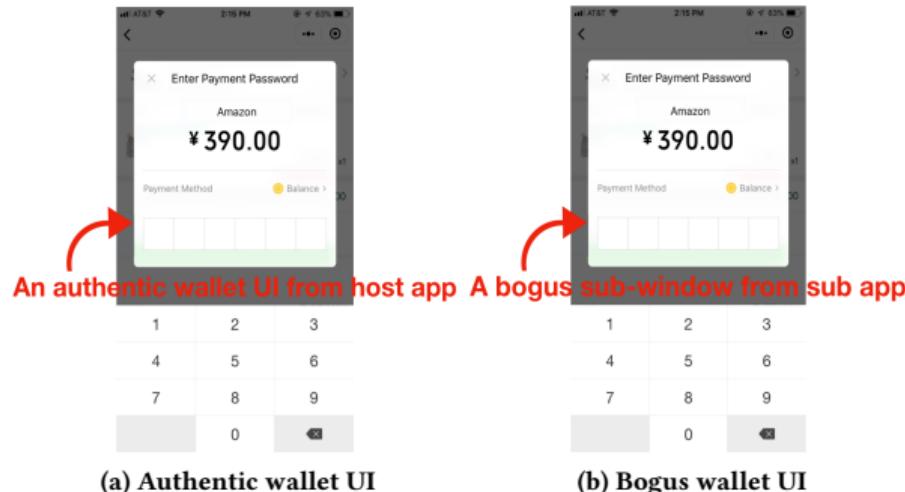


Online Earning



T8: Financial Charge Malware

Financial Charge via Phishing Malware



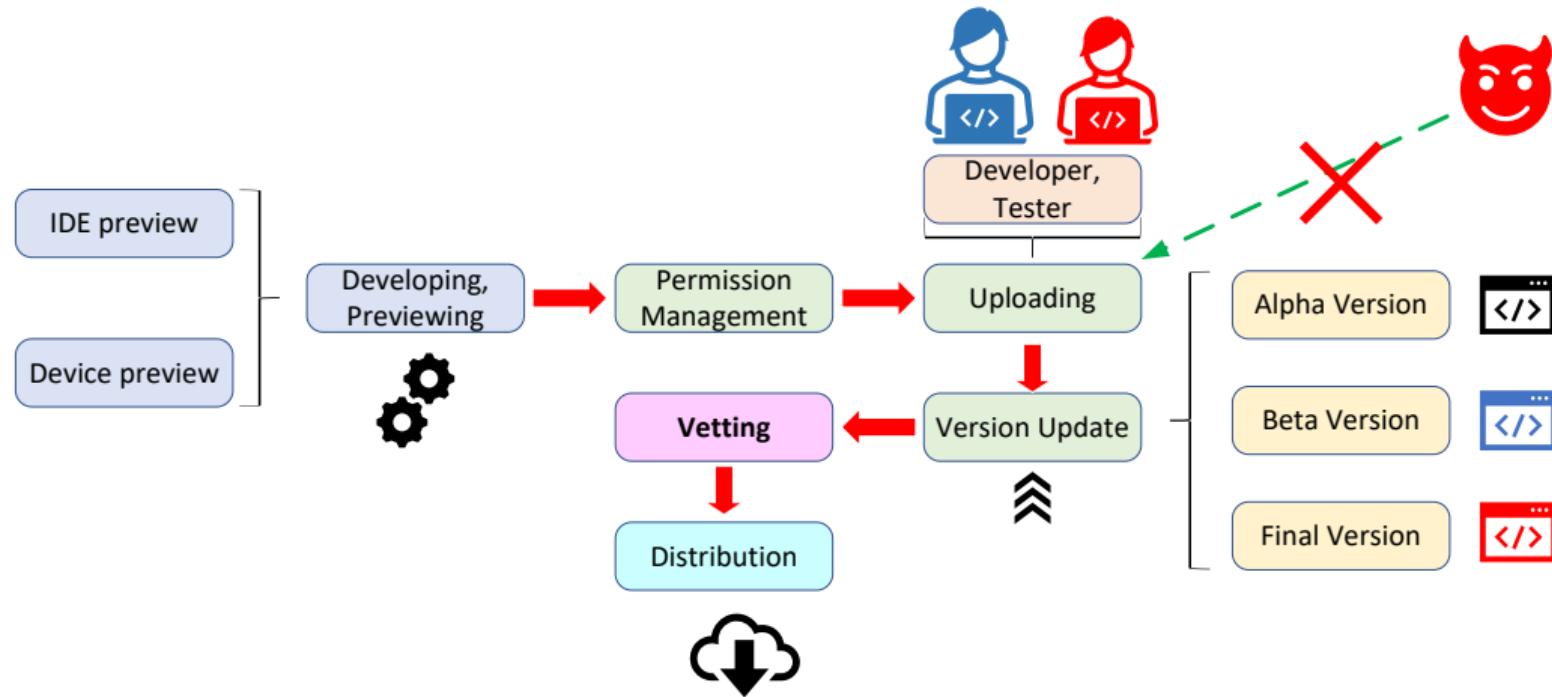
Mobile wallet UI confusion [LXX+20]

T9: Grayware

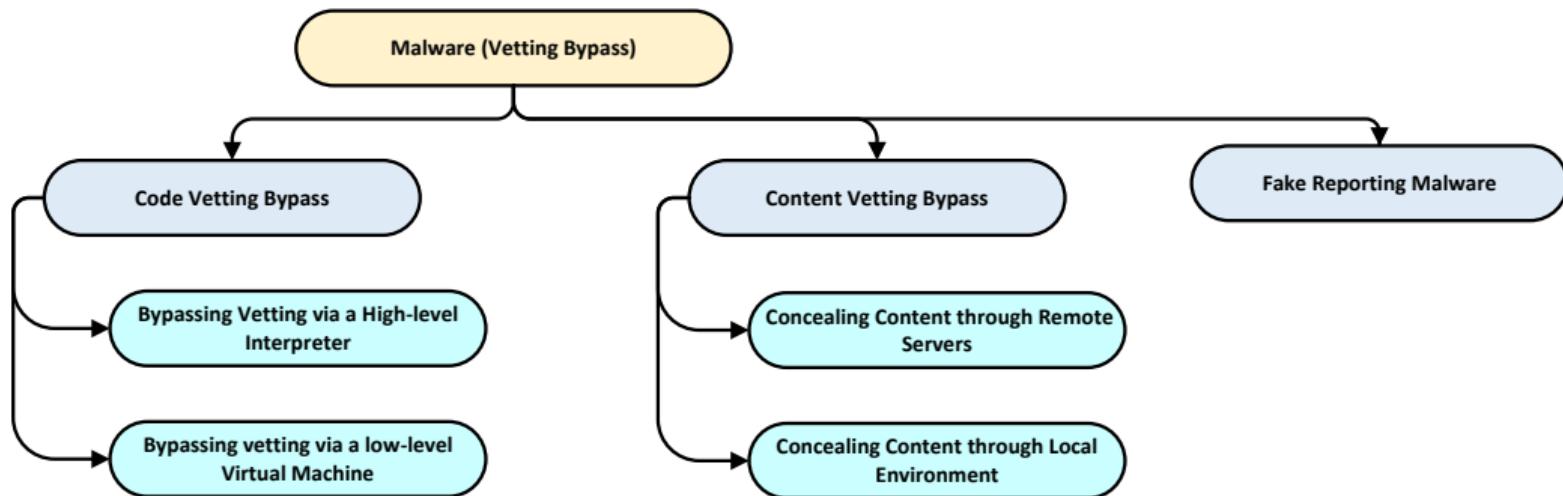
Gambling [opr]



Code Vetting



Vetting Bypassing Malware



T10: Code Vetting Bypassing Malware

Bypassing Vetting via Interpreters [CN]

README.md

mini-hot ↗

⚠ 注意：该方案使用的开源库已被微信官方禁用，谨慎使用！

npm v0.2.4

[Demo 工程](#)

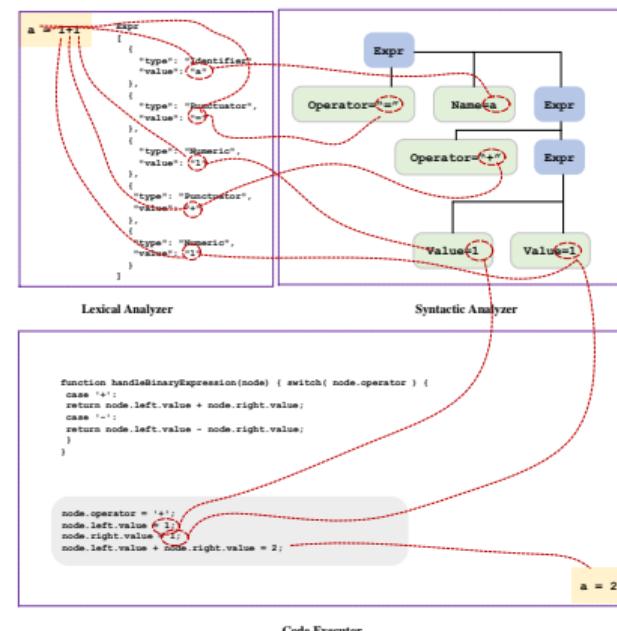
API ↗

createRemotePage - 单个页面远程加载 ↗

```
// SomePage.ts
import { createRemotePage } from '@mini-hot/taro'
export default createRemotePage(() => import('./SomePage'))
```

createRemoteApp - 小程序 SPA 化后远程加载 ↗

```
// SPA.ts
import { createRemoteApp } from '@mini-hot/taro'
```



T11: Content Vetting Bypassing Malware [Sin]



```
<!--pages/add/add.wxml-->
<view wx:if="{{ischeck}}" class="p">
    添加功能尚在完善中
    <view class="in">
        这是一个英语句子随机小程序，给英语爱好者提供丰富的英语美剧供欣赏或摘抄。
    </view>
</view>
<!-- 给审核看的页面 -->

<view wx:else class="box">...
</view>
<!-- 给用户看的页面 -->
```



Controlled By Time

```
var time='2021-03-24 21:17:15';//是三天后的时间
var t= util.fulltime(new Date());//返回现在的时间
this.setData([
  ischeck: t<time?true:false,
  //现在的时间大于三天后的时间是false，表示没有在审核，正常显示页面
]);
```



Controlled By Remote Server

```
1 | onLaunch (options) {
2 |   const accountInfo = wx.getAccountInfoSync();
3 |   if(accountInfo.miniProgram.envVersion === 'develop'){
4 |     // 跳转预留好的页面
5 |     wx.navigateTo() // 开发时，可以注释本行，方便自己预览
6 |   }
7 | },
```

T12: Reporting Bypassing Malware

5:17 21% Weixin Report Against FAKE One Hit 999 Pornographic vulgarity Enter content to report 0/200 Screenshots of Evidence(1/4)



Enter content to report 0/200 Screenshots of Evidence(1/4)

5:16 22% Weixin Report Against Authentic Frostwing Message Board Pornographic vulgarity Enter content to report 0/200 Screenshots of Evidence(1/4)



Enter content to report 0/200 Screenshots of Evidence(1/4)

Allow WeChat to use data and screenshots
of the current page of the Mini Program as
evidence of complaints.[Related Notes](#)

Submit

Allow WeChat to use data and screenshots
of the current page of the Mini Program as
evidence of complaints.[Related Notes](#)

Submit

T12: Reporting Bypassing Malware

5:17 5:16

Weixin Weixin

Report Against Report Against

One Hit 999 Frostwing Message Board

Pornographic vulgarity Pornographic vulgarity

Enter content to report Enter content to report

0/200 0/200

Screenshots of Evidence(1/4) Screenshots of Evidence(1/4)

+ +

Allow WeChat to use data and screenshots
of the current page of the Mini Program as
evidence of complaints.[Related Notes](#)

Allow WeChat to use data and screenshots
of the current page of the Mini Program as
evidence of complaints.[Related Notes](#)

Submit Submit

	Web App	Mobile App	Miniapp
Environment	Browser	Mobile Operating System	Suer App
Authority	Decentralized	Seperate App Store	Super App
Vetting	Decentralized	By Certain App Store	By Super App
Reporting	Decentralized	Write email to App Store	Via built-in Inter.

Table: Comparison of the authorities

Outline

1 Introduction

2 Threats from Vulnerability Exploitation

3 Threats from Malware Attacks

4 Takeaway

The World of Mobile Super Apps (“One App with Multiple Services”)



Security Threats

Threats from Vulnerability Exploitation

① Vulnerabilities in Host Apps

- (T1) Platform Discrepancies [WZL23a]
- (T2) Privileged APIs [WZL23b]
- (T3) Identity Confusion [ZZL⁺²²]

② Vulnerabilities in Miniapps

- (T4) Cross Miniapp Request Forgery [YZL22]
- (T5) AppSecret Key Leakage [ZYL23]
- (T6) Missing Signature Verification [ZZW23]

Threats from Malware Attacks

① API Misuse/Abuse (Payload)

- (T7) Collecting User Privacy
- (T8) Service Abusing
- (T9) Grayware

② Bypassing Vetting

- (T10) Code Vetting Bypassing
- (T11) Content Vetting Bypassing
- (T12) Reporting Bypassing

Other Open Problems

Vulnerability Identification

- ▶ Memory vulnerabilities (e.g., JavaScript engines, native layers)
- ▶ Logic vulnerabilities in both host apps (e.g., permission mgmt) and miniapps

Security/Privacy Compliance Analysis

- ▶ Various regulations/laws in privacy-rich platform
- ▶ Tools for compliance checks, and even supply chain analysis

Malware Analysis

- ▶ Semantic-aware miniapp vetting
- ▶ Developing static, dynamic, or symbolic analysis tools for miniapp malware analysis

Security Mechanism Standardization

- ▶ Super app implementation variations can cause security risks.
- ▶ Standardizing the interface/APIs for these platforms.

Thank You

Unpacking the Threats of All-in-One Mobile Super Apps

Zhiqiang Lin

zlin@cse.ohio-state.edu

October 17th, 2023

References |

-  Wu Changming and Super Nos, *mini-hot*, <https://github.com/mini-hot/mini-hot>.
-  Haoran Lu, Luyi Xing, Yue Xiao, Yifan Zhang, Xiaojing Liao, XiaoFeng Wang, and Xueqiang Wang, *Demystifying resource management risks in emerging mobile app-in-app ecosystems*, Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, 2020, pp. 569–585.
-  Weixin mini program platform operation rules, <https://developers.weixin.qq.com/miniprogram/en/product/>.
-  SinkingPeople, *How to bypass vetting?*, https://blog.csdn.net/weixin_43614065/article/details/125778486.
-  Chao Wang, Ronny Ko, Yue Zhang, Yuqing Yang, and Zhiqiang Lin, *Taintmini: Detecting flow of sensitive data in mini-programs with static taint analysis*, ICSE.
-  Chao Wang, Yue Zhang, and Zhiqiang Lin, *One size does not fit all: Uncovering and exploiting cross platform discrepant apis in wechat*, 31st USENIX Security Symposium (USENIX Security 23), 2023.
-  _____, *Uncovering and exploiting hidden apis in mobile super apps*, Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, 2023.
-  Yuqing Yang, Yue Zhang, and Zhiqiang Lin, *Cross miniapp request forgery: Root causes, attacks, and vulnerability detection*, Proceedings of the 29th ACM Conference on Computer and Communications Security, 2022.
-  Yue Zhang, Bayan Turkistani, Allen Yuqing Yang, Chaoshun Zuo, and Zhiqiang Lin, *A measurement study of wechat mini-apps*, Proceedings of the ACM on Measurement and Analysis of Computing Systems 5 (2021), no. 2, 1–25.

References II

-  Yue Zhang, Yuqing Yang, and Zhiqiang Lin, *Don't leak your keys: Understanding, measuring, and exploiting the appsecret leaks in mini-programs.*, Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, 2023.
-  Lei Zhang, Zhibo Zhang, Ancong Liu, Yinzhi Cao, Xiaohan Zhang, Yanjun Chen, Yuan Zhang, Guangliang Yang, and Min Yang, *Identity confusion in webview-based mobile app-in-app ecosystems*, 31st USENIX Security Symposium (USENIX Security'22), 2022.
-  Yanjie Zhao, Yue Zhang, and Haoyu Wang, *Potential risks arising from the absence of signature verification in miniapp plugins*, ACM Workshop on Secure and Trustworthy Superapps (SaTS), 2023.