Your secret files are mine:
Bug finding and exploit techniques on file transfer app of all top Android vendors

Zhang Xiangqian
Liu Huiming

腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

About us



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

- **Tencent**
  - Largest social media and entertainment company in China
- **Tencent Security Xuanwu Lab**
  - Applied and real world security research
- **Members of Advanced Security Team**
  - Zhang Xiangqian(@h3rb0)
  - Huiming Liu (@liuhm09)

Contents



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

- Introduction
- Attack Surfaces and Previous Works
- Vulnerabilities & Exploit & Demo
- Mitigation
- Conclusion

Introduction



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

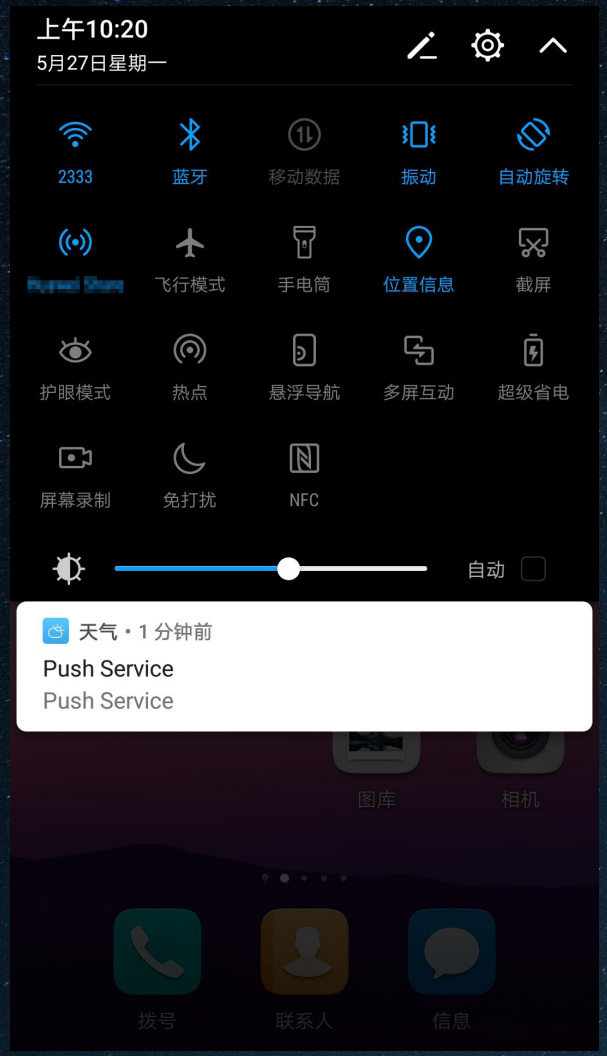
- Nearby Transmission Technologies
  - Bluetooth, Android beam, WiFi hotpot, WiFi Direct
- Nearby Sharing on Android
  - Vendor’s App
    - xxxx Share, xxxx Drop …
  - Third Party App
    - SHAREit, Xender, AirDroid, Send Anywhere ……

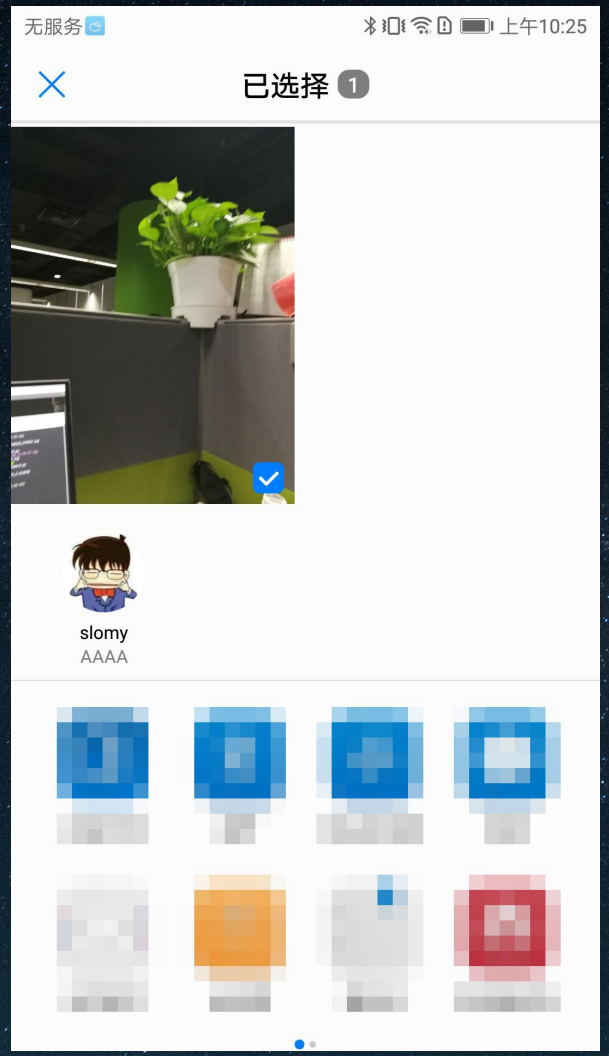
Motivation

- Nearby file transfer
  - Bluetooth/NFC(Android Beam)
    - Not user friendly enough
    - Not fast enough
  - APPs based on Wi-Fi related Technology
- Nearly all top Android Vendors have their Nearby file transfer APPs
  - 2018 shipment: nearly a billion android devices (IDC)
- Third-party file transfer Apps
  - Top 10 Apps: nearly a billion users
- But are those Apps safe?

Android Sharing Apps

腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

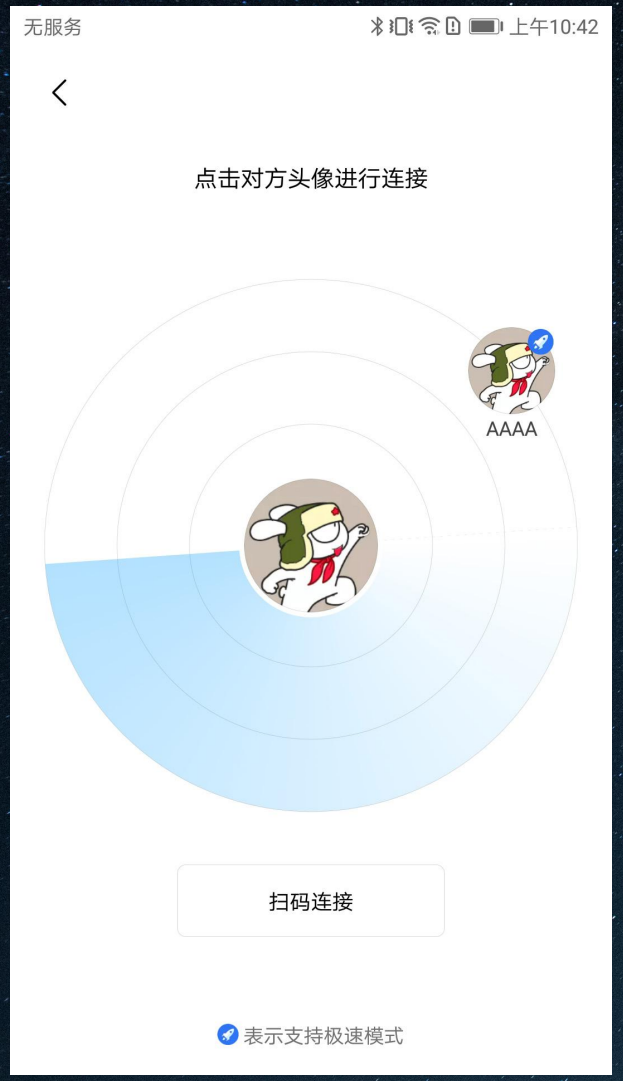




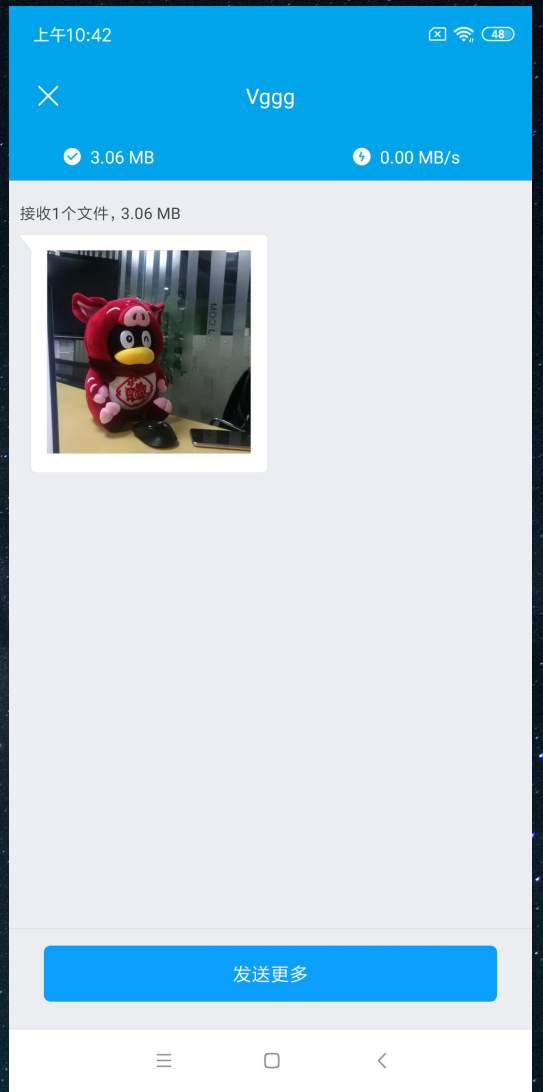


Android Sharing Apps

腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB







Android Sharing Apps

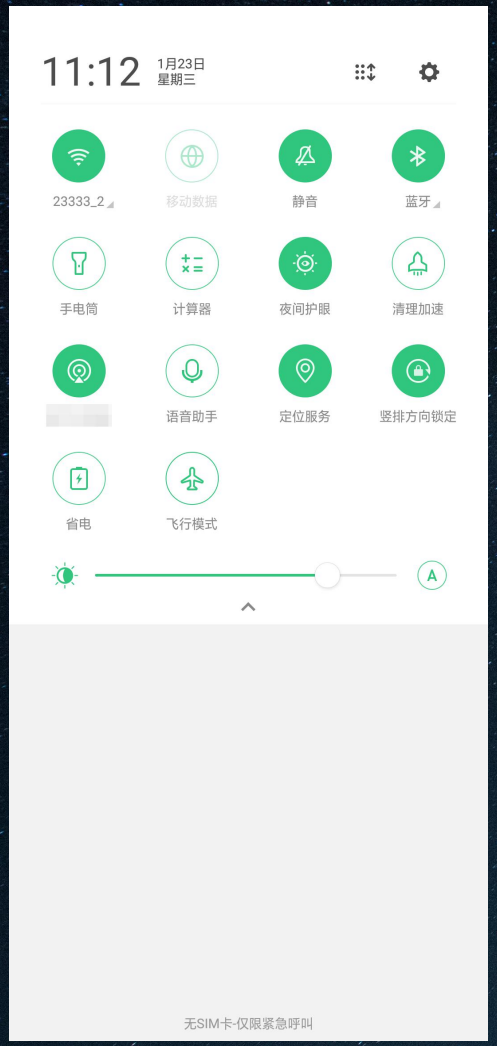


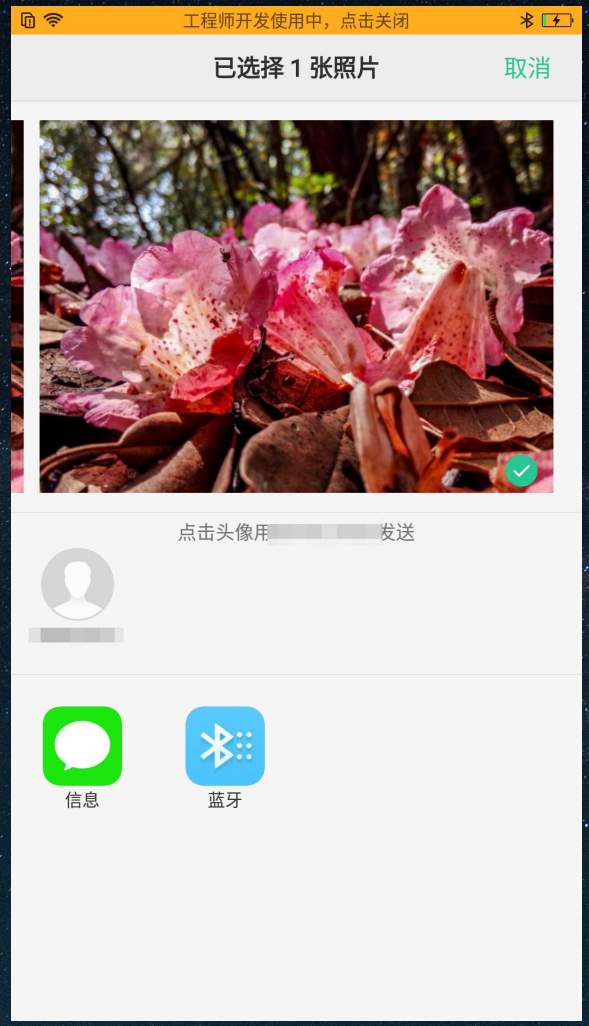
腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

- Scan The QR code to connect
- Shake at the same time
- …

Android Sharing Apps

腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB







Sharing Process



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

Discover	• BLE advertising Name, ID, Device Type ...
Pair	• Automatic pairing and key exchange
Connect	• WiFi / WiFi P2P
Transfer	• Pictures, Apks, Other Files ……

Attack surface



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

- Adversary:
  - Attackers can be sniffer and sender of BT and Wi-Fi packets
- Damage:
  - Attackers can get the transferred files
  - Others: Traffic hijacking, Info leaking, RCE…
- Attack Surfaces:
  - Link Establishing
  - Secure Transmission
  - Device / ID Spoofing
  - MITM
  - Others(Web Server)

Link Establishing



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

- Connect automatically
  - Attackers may join into the network without users' permission.
- Networks' key negotiating
  - Attackers may obtain the established networks' passwords.

Secure transmission

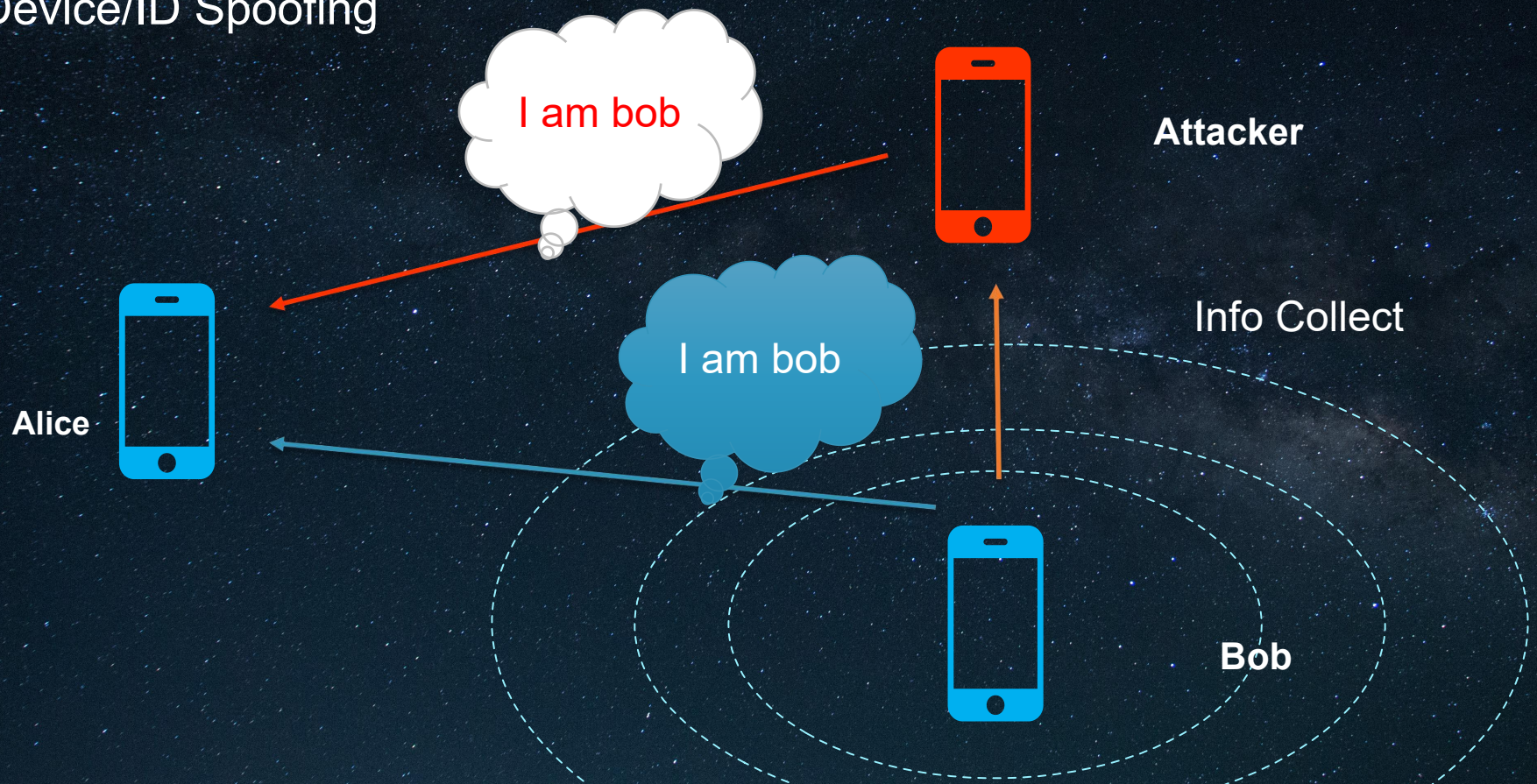


腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

- No Encryption
  - Get the file directly
- Unsafe key exchange
  - Recover the real file from the encrypted traffic

Device / ID Spoofing

- Authenticate the real devices / users
  - Device/ID Spoofing



Man-in-the-middle

- Bluetooth MITM attack
- WiFi/WiFi P2P MITM attack

- Two Level
  - Users may feel something strange
  - The users can’t feel anything (✓)

Others



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

- Web Server
  - Directory traversal
- Other information disclosure
- Android components related

Previous researches



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

- ZeroConf on OSX / iOS
  - Discovering and Exploiting Novel Security Vulnerabilities in Apple ZeroConf -- Xiaolong Bai & Luyi Xing (S&P 2016)
- Reverse Google Nearby Connection API
  - Nearby Threats: Reversing, Analyzing, and Attacking Google's 'Nearby Connections' on Android – Antonioli, Daniele, etc. (NDSS '19)

Our Research



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

- First comprehensive research about Android Nearby Sharing Apps
- Analyzed many related Apps including Pre-installed / Third Party Apps
- Found a lot vulnerabilities in various Apps
- Arrange them in several common attack methods

Vulnerabilities

Vulnerabilities' categories



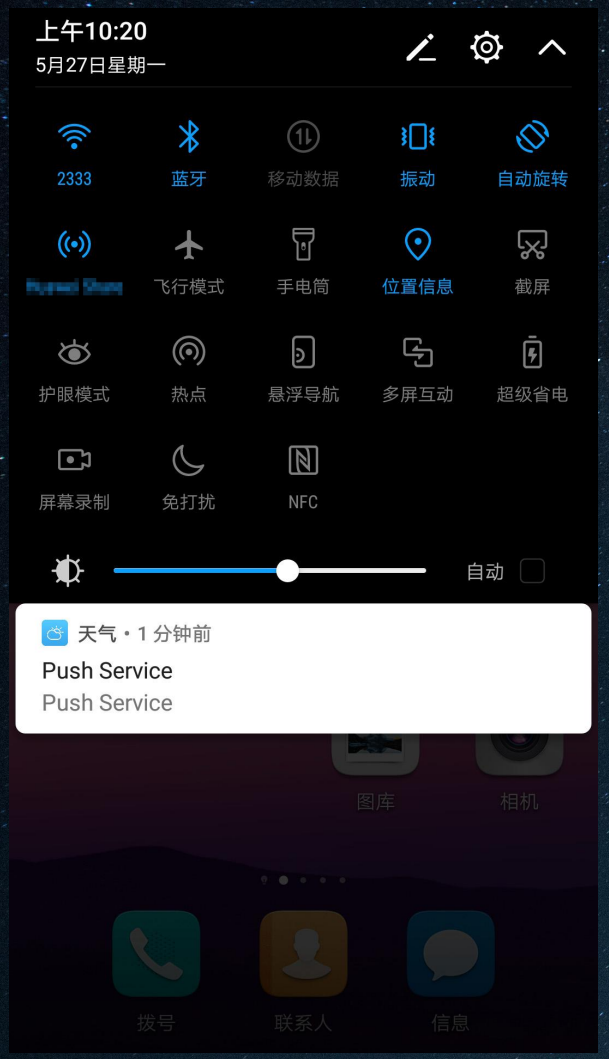
腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

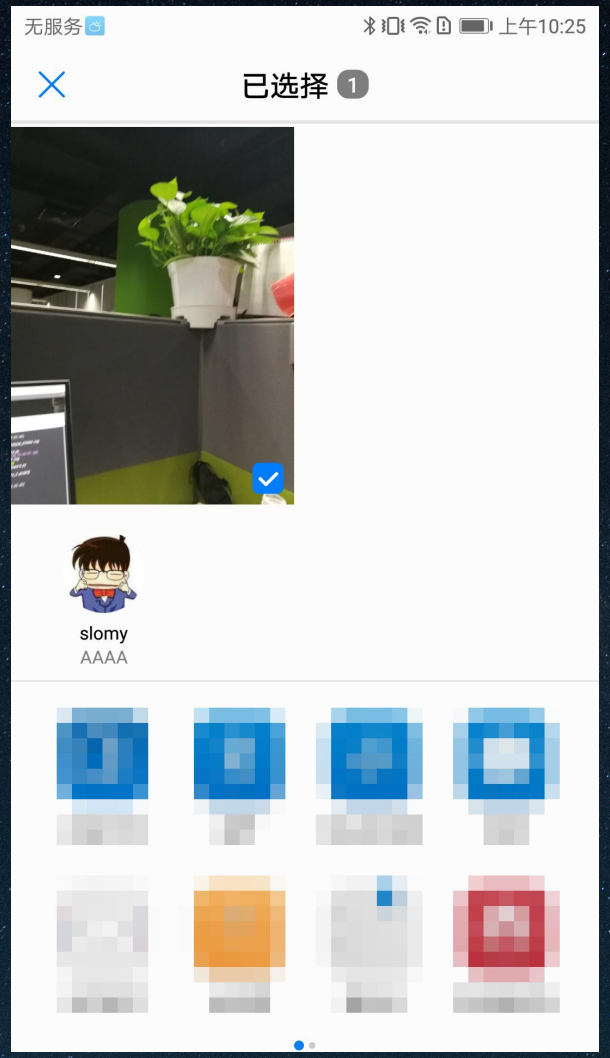
- Sniffer attack related vulnerabilities
- Man-in-the-middle attack related vulnerabilities
- Logic vulnerabilities
- Other vulnerabilities

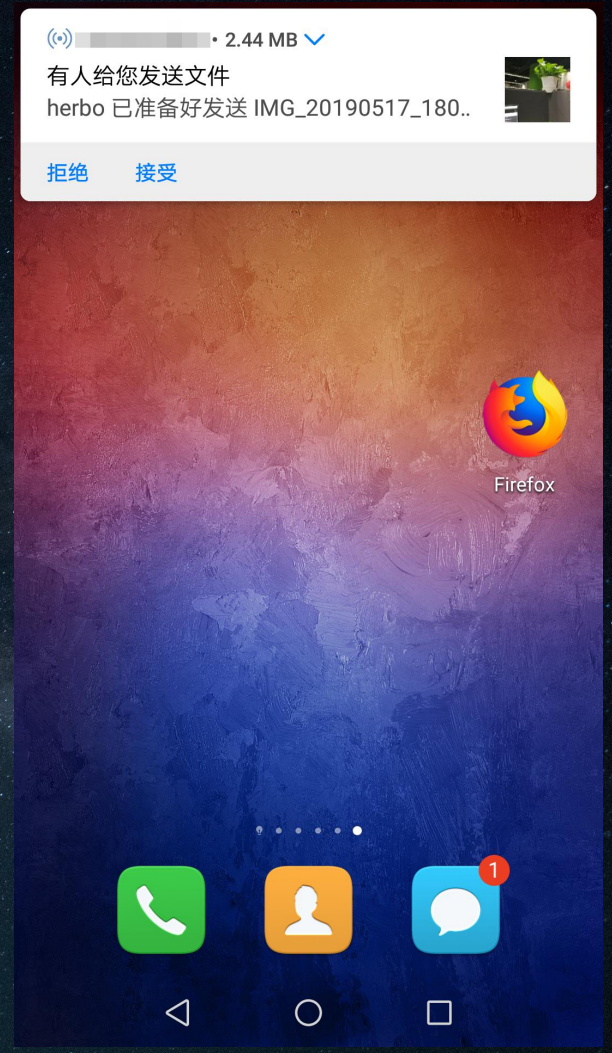
Sniffer attack

Sniffer attack

腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

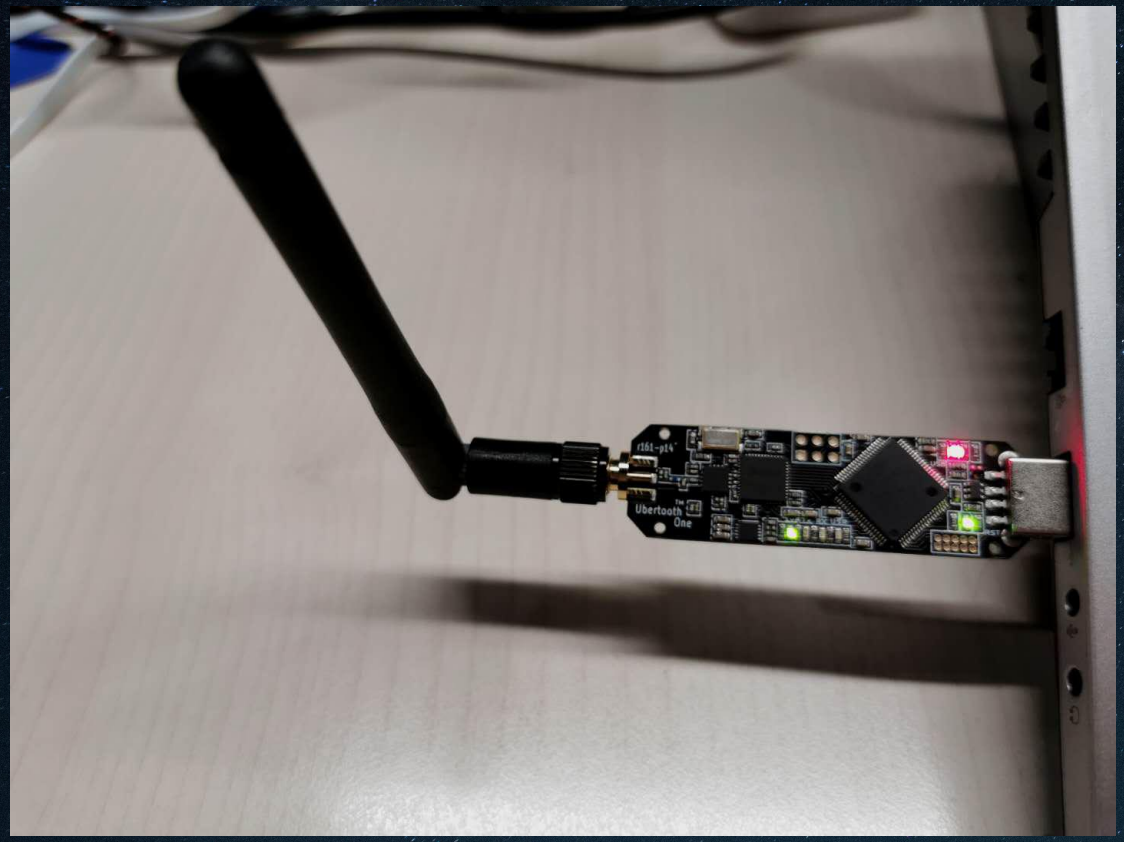






BLE Sniffer

- Ubertooth 、CC2540、nRF51



BLE Advertising


```
6229 119.380240 62:76:43:3b:f7:9a  <broadcast>   BLE ADV   33 SCAN_RSP
6333 120.628977 41:9c:b0:b1:59:7b  <broadcast>   BLE ADV   68 SCAN_RSP
6338 120.693979 6f:a4:95:57:2d:9c  <broadcast>   BLE ADV   68 SCAN_RSP
6343 120.710227 41:9c:b0:b1:59:7b  <broadcast>   BLE ADV   68 SCAN_RSP
6422 121.784448 5c:14:d2:ac:ff:30  <broadcast>   BLE ADV   39 SCAN_RSP
6526 123.487728 41:9c:b0:b1:59:7b  <broadcast>   BLE ADV   68 SCAN_RSP
```

```
⊞ Frame 6333: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface 0
⊞ PPI version 0, 24 bytes
  DLT: 147, Payload: btle (Bluetooth Low Energy)
⊟ Bluetooth Low Energy
    Access Address: 0x8e89bed6
  ⊞ Packet Header
    Advertising Address: 41:9c:b0:b1:59:7b (41:9c:b0:b1:59:7b)
  ⊟ Scan Response Data: 1c1635fe0f77485541574549204d617465203230205072 6f...
    ⊟ service data: 35fe0f77485541574549204d617465203230205072 6f2028...
        length: 0x1c
        type: Service Data for 16 bit UUID. (0x16)
        16 bit uuid: 0xfe35
        service data value: 0f77485541574549204d617465203230205072 6f20285544...
    CRC: 0x43fa0f
```

```
0000  00 00 18 00 93 00 00 00  36 75 0c 00 00 62 09 00   ........ 6u...b..
0010  77 20 e7 47 f1 df 56 00  d6 be 89 8e 44 23 7b 59   w .G..V. ....D#{Y
0020  b1 b0 9c 41 1c 16 35 fe  0f 77 48 55 41 57 45 49   ...A..5. .wHUAWEI
0030  20 4d 61 74 65 20 32 30  20 50 72 6f 20 28 55 44    Mate 20  Pro (UD
0040  29 43 fa 0f                                       )C..
```

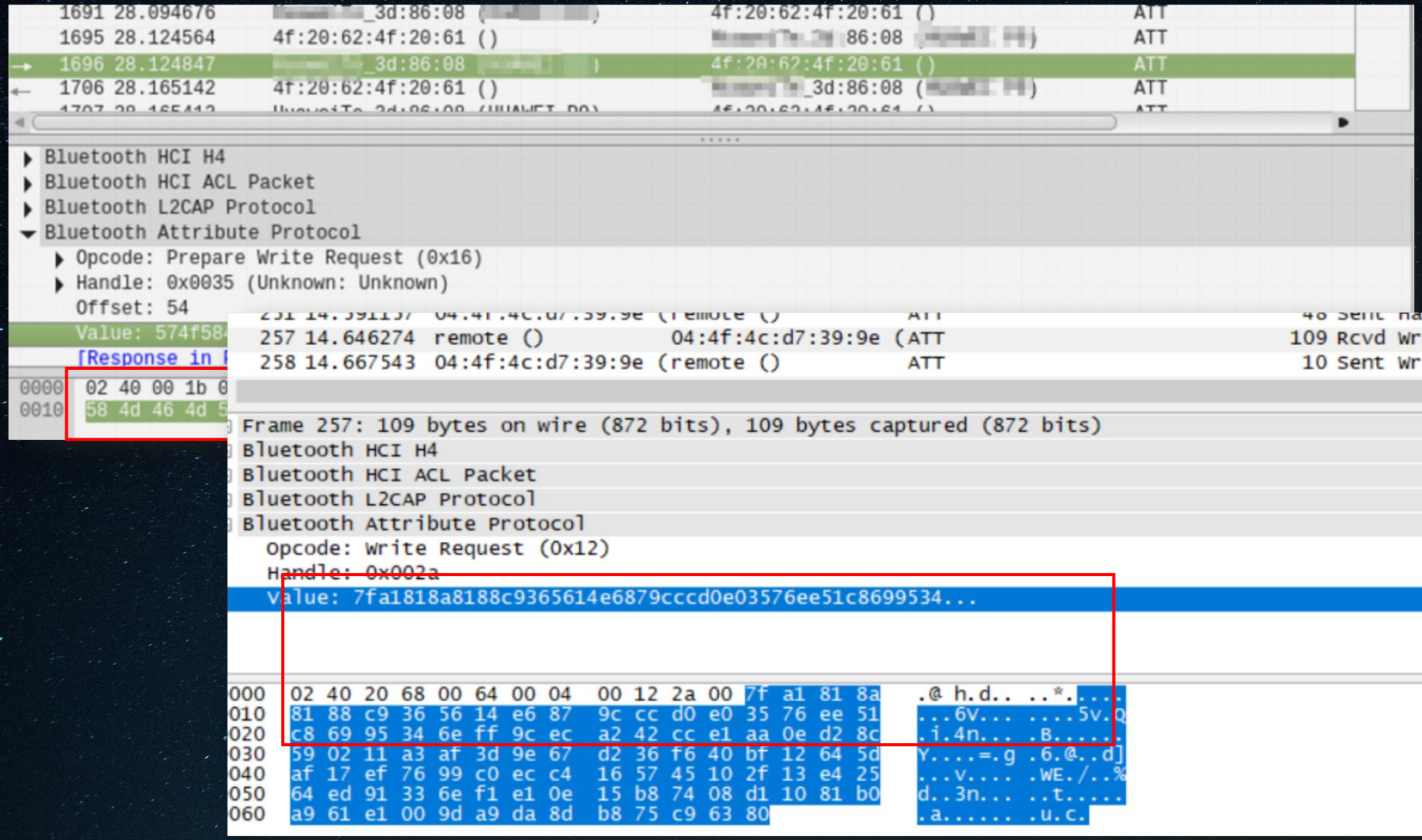


slomy

AAAA

Secret data

腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB



Secret data

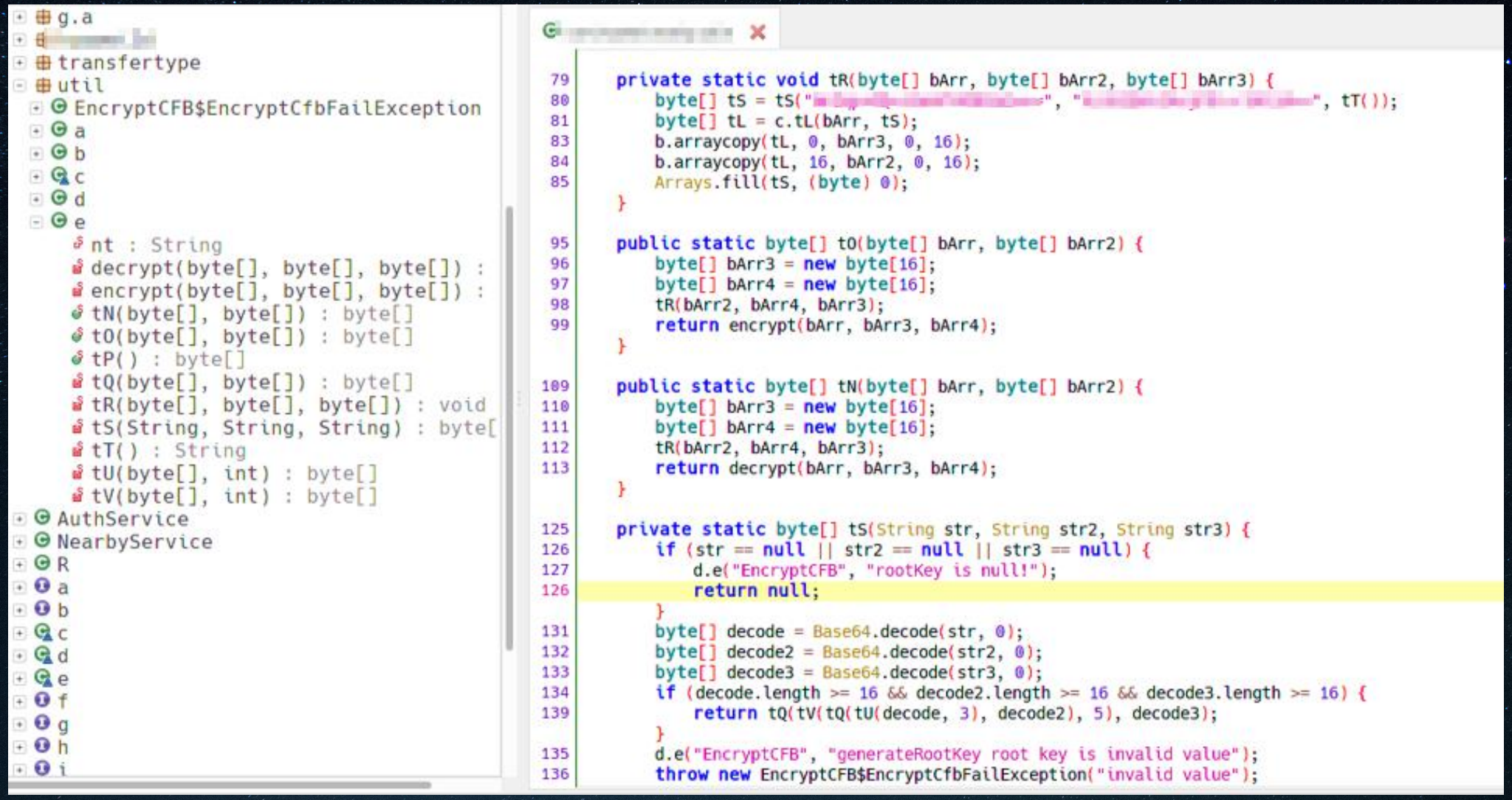


腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

- What does this secret data mean?

- 7fa1818a8188c9365614e6879cccd0e03576ee51c86995346eff9ceca242cce1aa0ed28c590211a3af3d9e67d236f640bf12645daf17ef7699c0ecc4165745102f13e42564ed91336ef1e10e15b87408d11081b0a961e1009da9da8db875c9638000000094000000

Reverse


g.a
transfertype
util
EncryptCFB$EncryptCfbFailException
a
b
c
d
e
nt : String
decrypt(byte[], byte[], byte[]) :
encrypt(byte[], byte[], byte[]) :
tN(byte[], byte[]) : byte[]
tO(byte[], byte[]) : byte[]
tP() : byte[]
tQ(byte[], byte[]) : byte[]
tR(byte[], byte[], byte[]) : void
tS(String, String, String) : byte[
tT() : String
tU(byte[], int) : byte[]
tV(byte[], int) : byte[]
AuthService
NearbyService
R
a
b
c
d
e
f
g
h
i
79 private static void tR(byte[] bArr, byte[] bArr2, byte[] bArr3) {
80 byte[] tS = tS("
tT());
81 byte[] tL = c.tL(bArr, tS);
83 b.arraycopy(tL, 0, bArr3, 0, 16);
84 b.arraycopy(tL, 16, bArr2, 0, 16);
85 Arrays.fill(tS, (byte) 0);
95 public static byte[] tO(byte[] bArr, byte[] bArr2) {
96 byte[] bArr3 = new byte[16];
97 byte[] bArr4 = new byte[16];
98 tR(bArr2, bArr4, bArr3);
99 return encrypt(bArr, bArr3, bArr4);
109 public static byte[] tN(byte[] bArr, byte[] bArr2) {
110 byte[] bArr3 = new byte[16];
111 byte[] bArr4 = new byte[16];
112 tR(bArr2, bArr4, bArr3);
113 return decrypt(bArr, bArr3, bArr4);
125 private static byte[] tS(String str, String str2, String str3) {
126 if (str == null || str2 == null || str3 == null) {
127 d.e("EncryptCFB", "rootKey is null!");
126 return null;
131 byte[] decode = Base64.decode(str, 0);
132 byte[] decode2 = Base64.decode(str2, 0);
133 byte[] decode3 = Base64.decode(str3, 0);
134 if (decode.length >= 16 && decode2.length >= 16 && decode3.length >= 16) {
139 return tQ(tV(tQ(tU(decode, 3), decode2), 5), decode3);
135 d.e("EncryptCFB", "generateRootKey root key is invalid value");
136 throw new EncryptCFB$EncryptCfbFailException("invalid value");

Older version



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

```
private String decryptP2PInfo(String p2pinfo) {
    if (p2pinfo == null) {
        MagicLinkLog.e("MagicLink-BtAndWifiWrapper", "decryptP2PInfo P2Pinfo is null.");
        return null;
    }
    String[] sourceStrArray = p2pinfo.split(":", 3);
    String p2pCfgEx = null;
    if (sourceStrArray.length != 3) {
        MagicLinkLog.e("MagicLink-BtAndWifiWrapper", "decryptP2PInfo P2Pinfo is invalid.");
        return null;
    }
    try {
        p2pCfgEx = Encrypt.decrypt(sourceStrArray[0], Encrypt.generateSecureKey(Encrypt.generateRootKey("[illegible]", "[illegible]", this.mSecureRandom), sourceStrArray[2]), sourceStrArray[1]);
    } catch (Exception e) {
        e.printStackTrace();
    }
    return p2pCfgEx;
}
```

Older version


```
public static String decrypt(String sSrc, String sKey, String sIv) {
    if (sSrc == null || sKey == null || sIv == null) {
        MagicLinkLog.e("BleEncrypt", "decrypt Key is null");
        return null;
    } else if (sKey.length() != 25) {
        MagicLinkLog.e("BleEncrypt", "decrypt - Length of Key is not 25byte" + sKey.length());
        return null;
    } else {
        byte[] byteKey = Base64.decode(sKey, 0);
        byte[] byteIv = Base64.decode(sIv, 0);
        byte[] encrypted = Base64.decode(sSrc, 0);
        SecretKeySpec sKeySpec = new SecretKeySpec(byteKey, "AES");
        try {
            Cipher cipher = Cipher.getInstance("[illegible]Padding");
            cipher.init(2, sKeySpec, new IvParameterSpec(byteIv));
            return new String(cipher.doFinal(encrypted), "UTF-8");
        } catch (RuntimeException e) {
            e.printStackTrace();
            return null;
        } catch (Exception e2) {
            e2.printStackTrace();
            return null;
        }
    }
}
```

Decrypt



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

```
p2pCfgEx = Encrypt.decrypt(sourceStrArray[0],
Encrypt.generateSecureKey(Encrypt.generateRootKey("XXXXXXXXXXXXXXX", "
XXXXXXXXXXXXXXX", this.mSecureRandom), sourceStrArray[2]), sourceStrArray[1]);
```

```
this.mSecureRandom =
service.getApplicationContext().getResources().getString(R.string.ble_xxxx_key);
```

/res/values/strings.xml:

```
<string name="ble_xxxx_key">XXXXXXXXXXXXXXX</string>
```

Decrypt

- Get **sourceStrArray** from BLE pcaket
- **sSrc** = sourceStrArray[0]
- **sIv** = sourceStrArray[1]
- **Rootkey** = enerateRootKey(" XXXXXXXXXXXXXXX ",
  " XXXXXXXXXXXXXXX ",mSecureRandom)
- **sKey** = generateSecureKey(rootkey,sourceStrArray[2])
- **p2p_info** = decrypt(sSrc,sKey,sIv)

Decrypt


```
$ python extract_password.py ble.pcap
rootkey:
sKey hex:
sKey:
sSrc:  cBPZzuOAoBJWUJ9/rqTOFsBYCHtBoGDMWGVBgiI+G5MO8CwKXhhol2ZvIIceClnV+g0C3OCaqKt+\nzUgWN2klMpEHmldlmhSRrrxicFaLBNQ=\n
sIv:  8HqTPrzdt9nuBzoMkV08/g==\n
P2Pinfo:
------------------------------
 id:  ('ygtat', 'BBBB')
 ip:  192.168.49.35
 ssid:  DIRECT-dZ-
 password:  xQTpdRN2
 version:  2462##versio
==============================
```

```
$ python new_extract_password.py ble.pcap
data 7fa1818a8188c9365614e6879cccd0e03576ee51c86995346eff9ceca242cce1aa0ed28c590
211a3af3d9e67d236f640bf12645daf17ef7699c0ecc4165745102f13e42564ed91336ef1e10e15b
87408d11081b0a961e1009da9da8db875c96380000000094000000
len: 208
Got p2p information!
P2Pinfo:
==============================
 ip:  192.168.49.2
 ssid:  DIRECT-ow-AAAA
 password:  DRtkSDs1
 version:  2412##version:3#
==============================
```

Sniffer



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

- Sniffer BLE
- Extract the secret data
- Decrypt the secret data
- Join the P2P Group
- ARP spoofing
- **Got the secret files!**

Demo 1



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

Demo 2



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

Man-in-the-middle attack

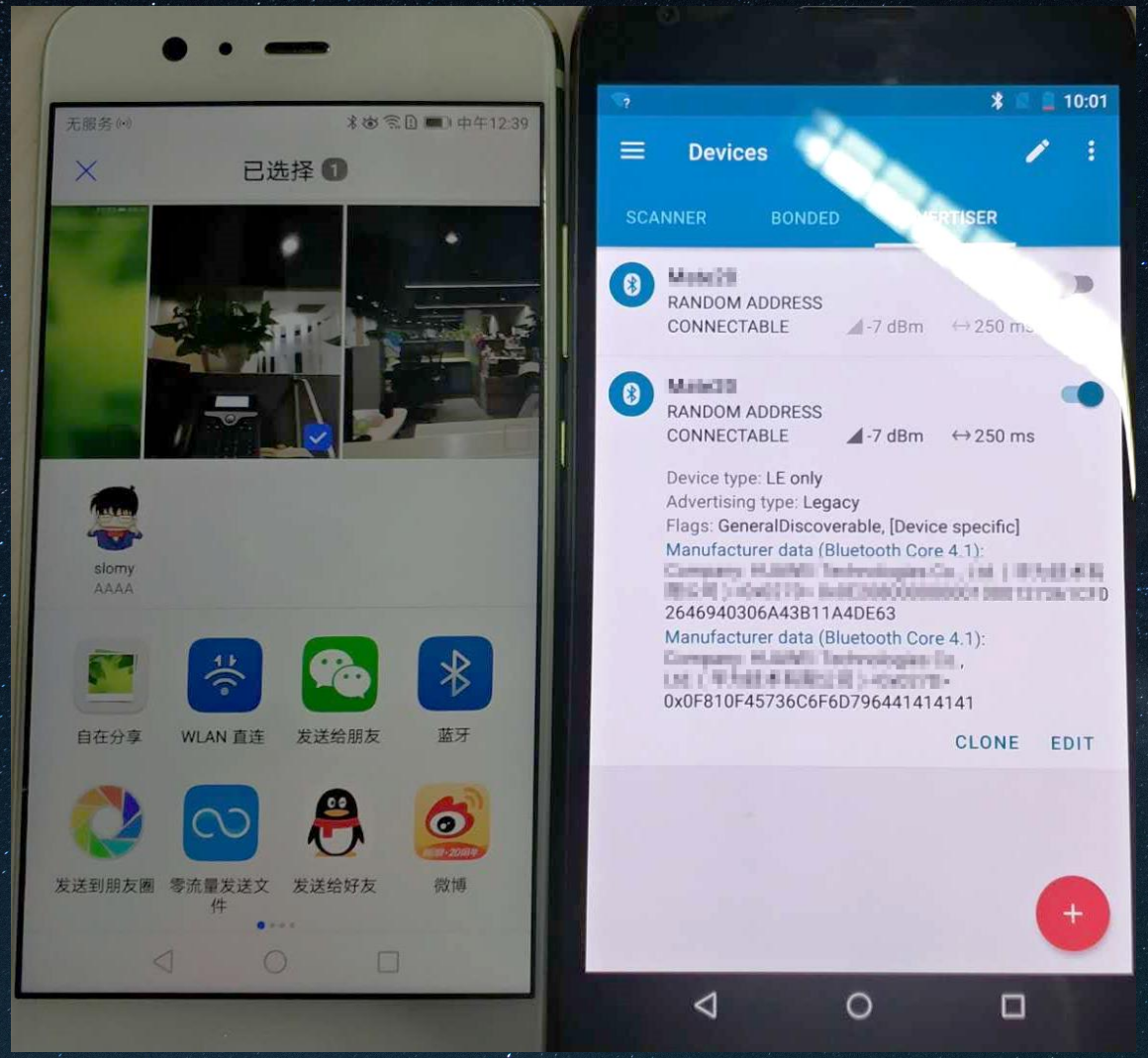
Device and ID spoofing

腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

- Who is **herbo zhang**?
- How to confirm identity?
- **DEVICE？ ID？ ICON？**



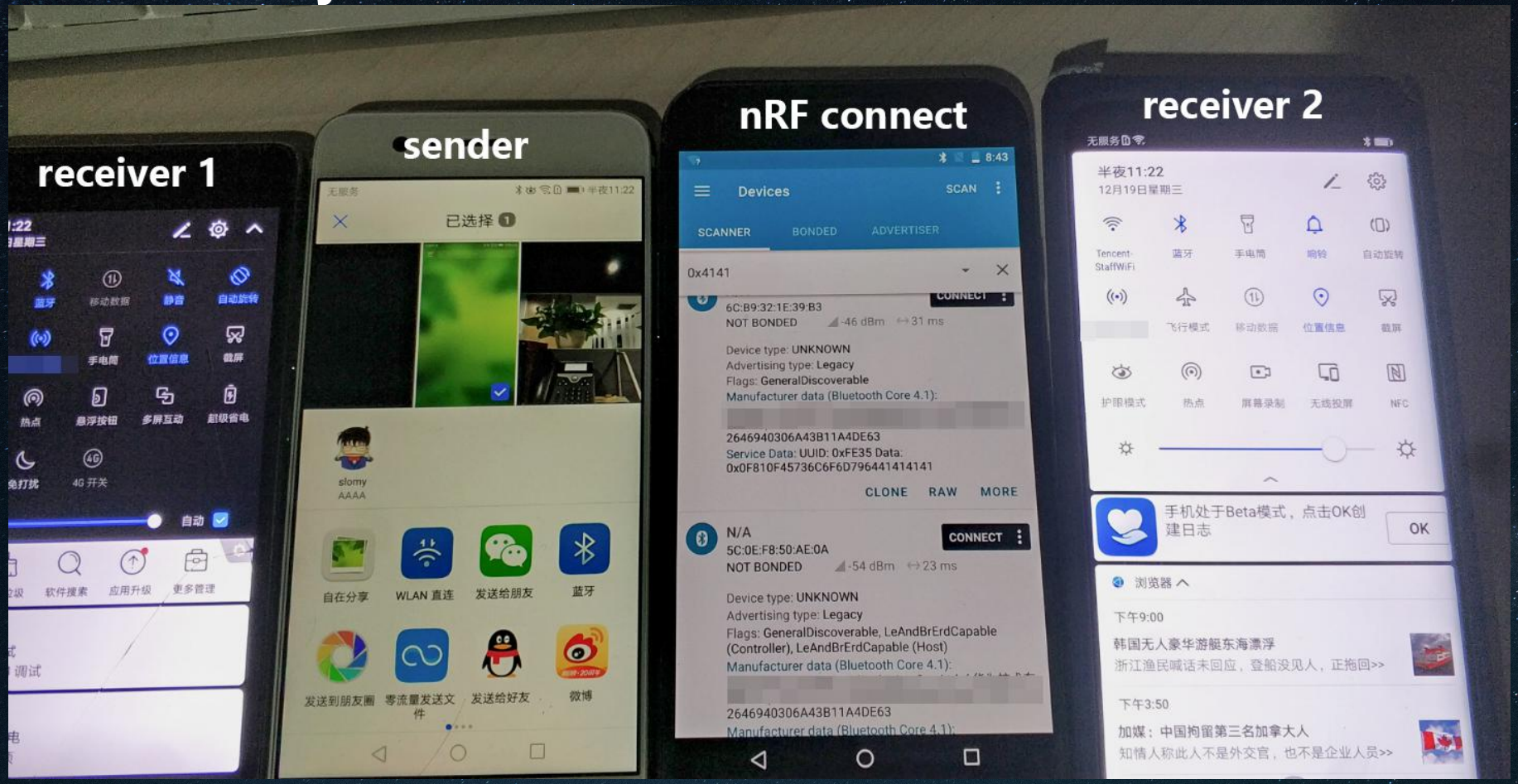
Device and ID spoofing



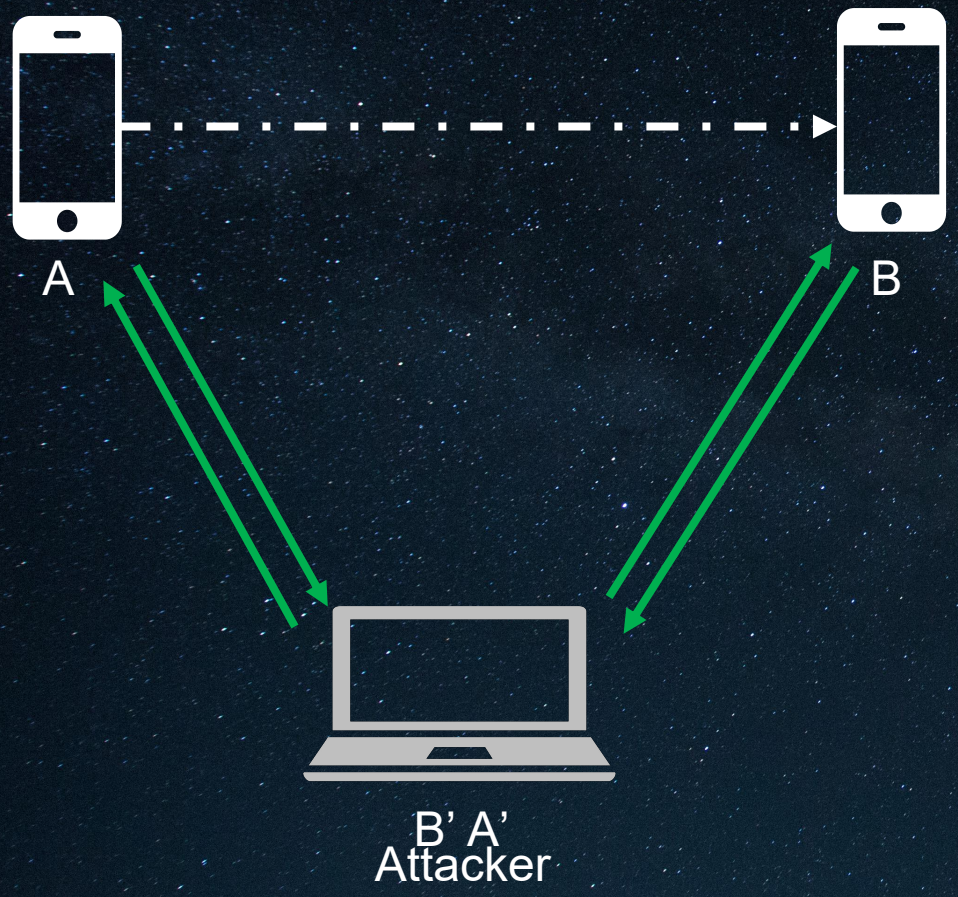
Device and spoofing

腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

- Who is **really** AAAA?



Man-in-the-middle



Who is the real receiver?

A has 50% chance to choose attacker.

Man-in-the-middle

- A can't find the attacker A’
  - The attacker does not send a scan response to A
- B can’t find the attack B’
  - B is not in discovery mode
- Attacker can distinguish between A and A', B and B’
  - The attacker can choose to block or receive any message
- The users (A,B) can’t feel anything

Demo 3



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

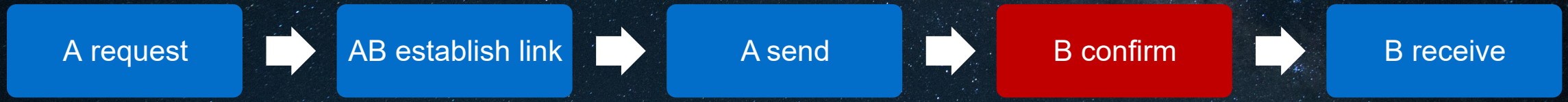
Logic vulnerabilities

None-confirm connection

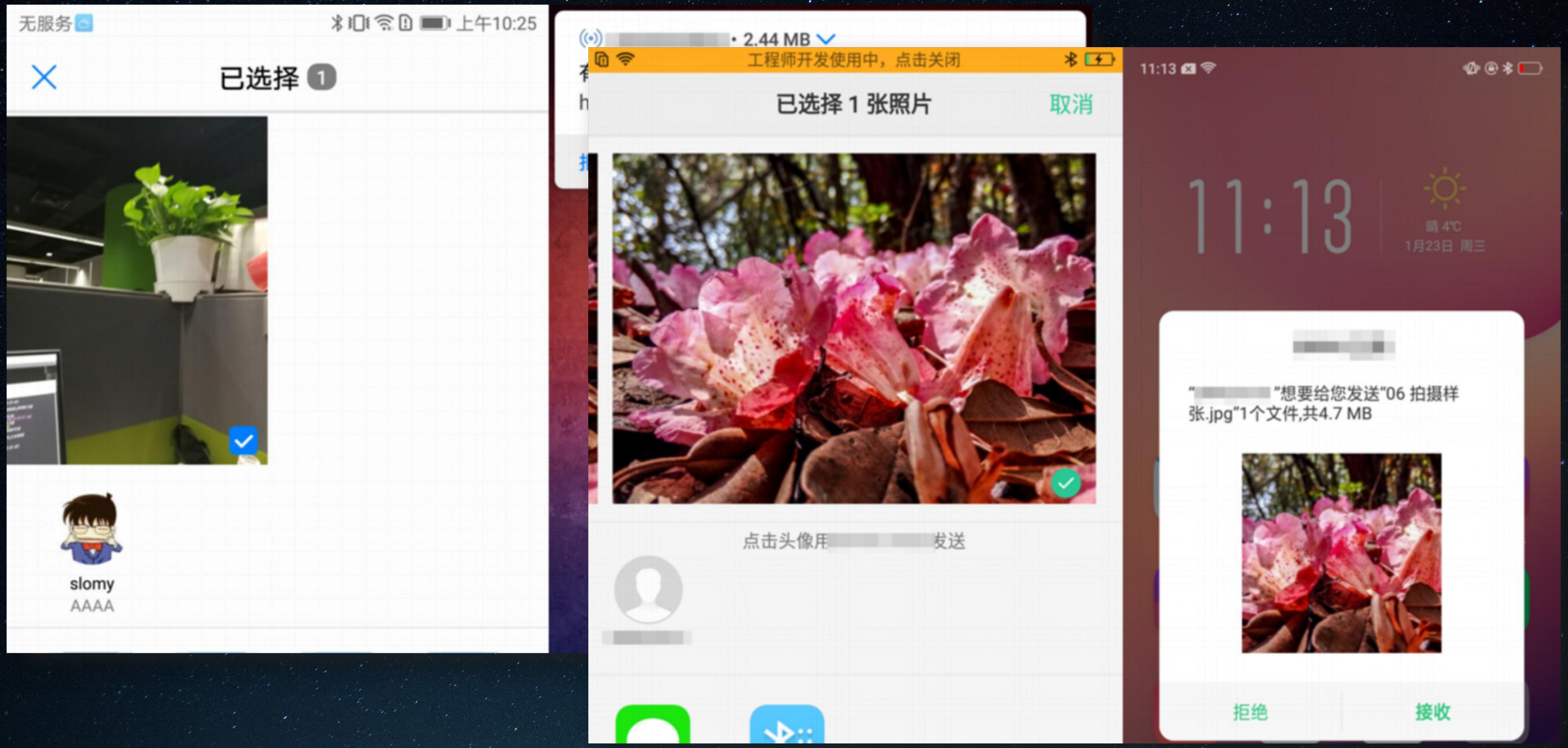
- Secure connection process



- Unsecure connection process



None-confirm connection



None-confirm connection



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

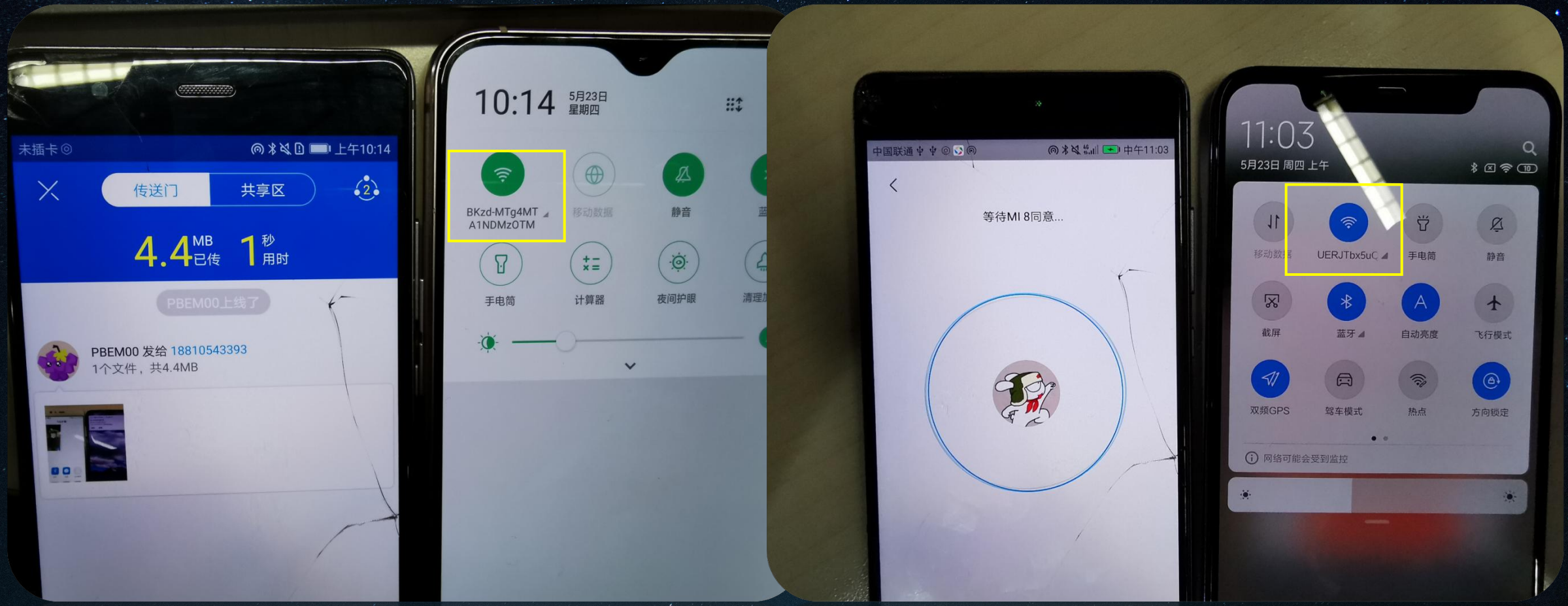
- Bring more attack surfaces

Zero click exploit?

None-confirm connection

腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

- Hijacking the network



Hijacking the network



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

- Who is server？ Sender？ Receiver？
- Can I choose to be a server? Yes!
- Can I make the other user connect to my WiFi without interaction？Yes!

Demo 4



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

Other vulnerabilities

Other vulnerabilities



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

- Accept automatically via Wi-Fi P2P
- Directory traversal
- RFM Vulnerabilities

Wi-Fi/Wi-Fi P2P Access Control

腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

- Need to input password



Accept automatically via Wi-Fi P2P

- However, when we use Wi-Fi P2P protocal to connect:
  - All authentications are gone!





Directory traversal



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

- A: Can I send you a file?
- B: Yes

- B: Please send it. I'm ready.
- A: I want to send it to you: /PATH/AAAA.JPG

- B: Send GET request for /PATH/AAAA.JPG
- A: Send /PATH/AAAA.JPG

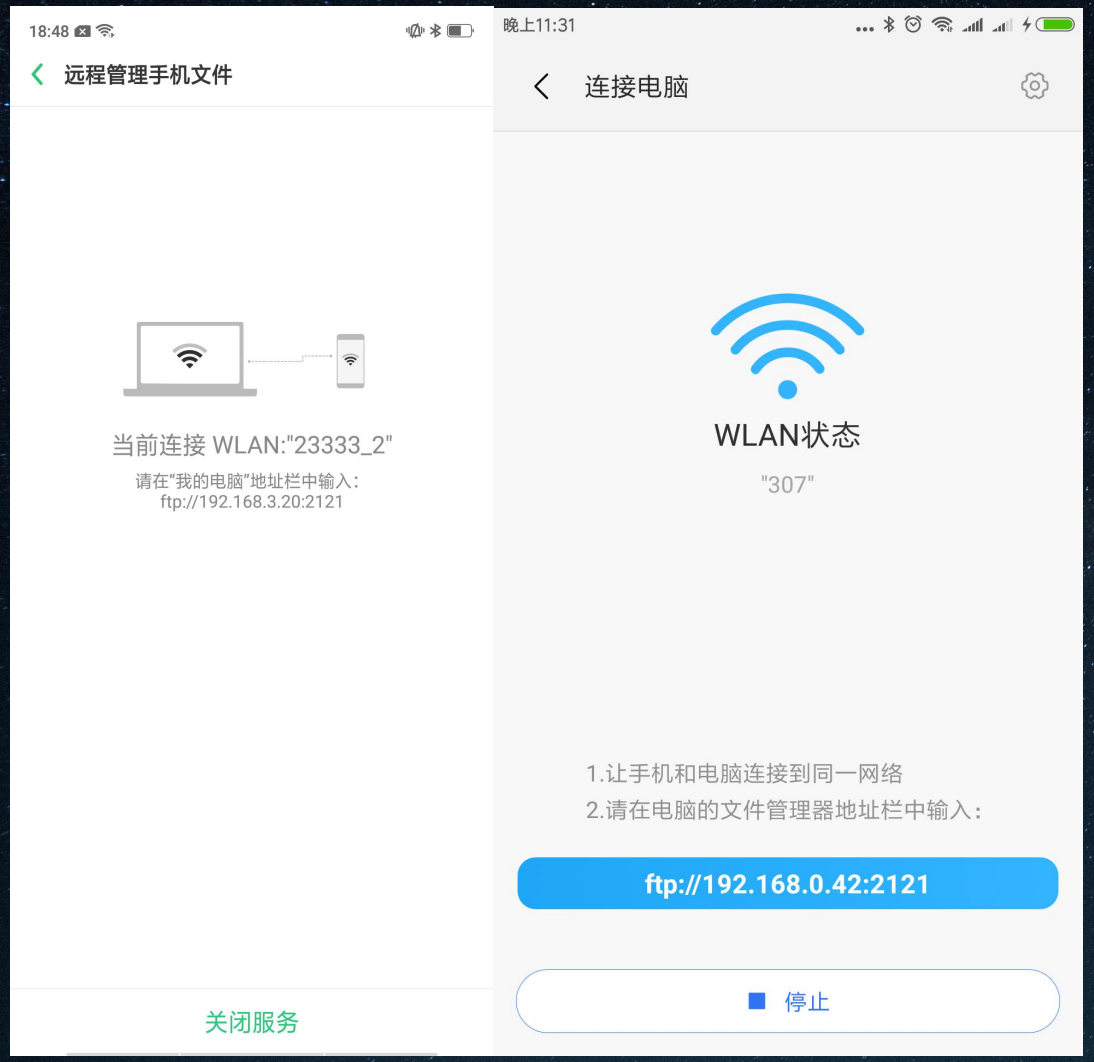
Demo 5



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

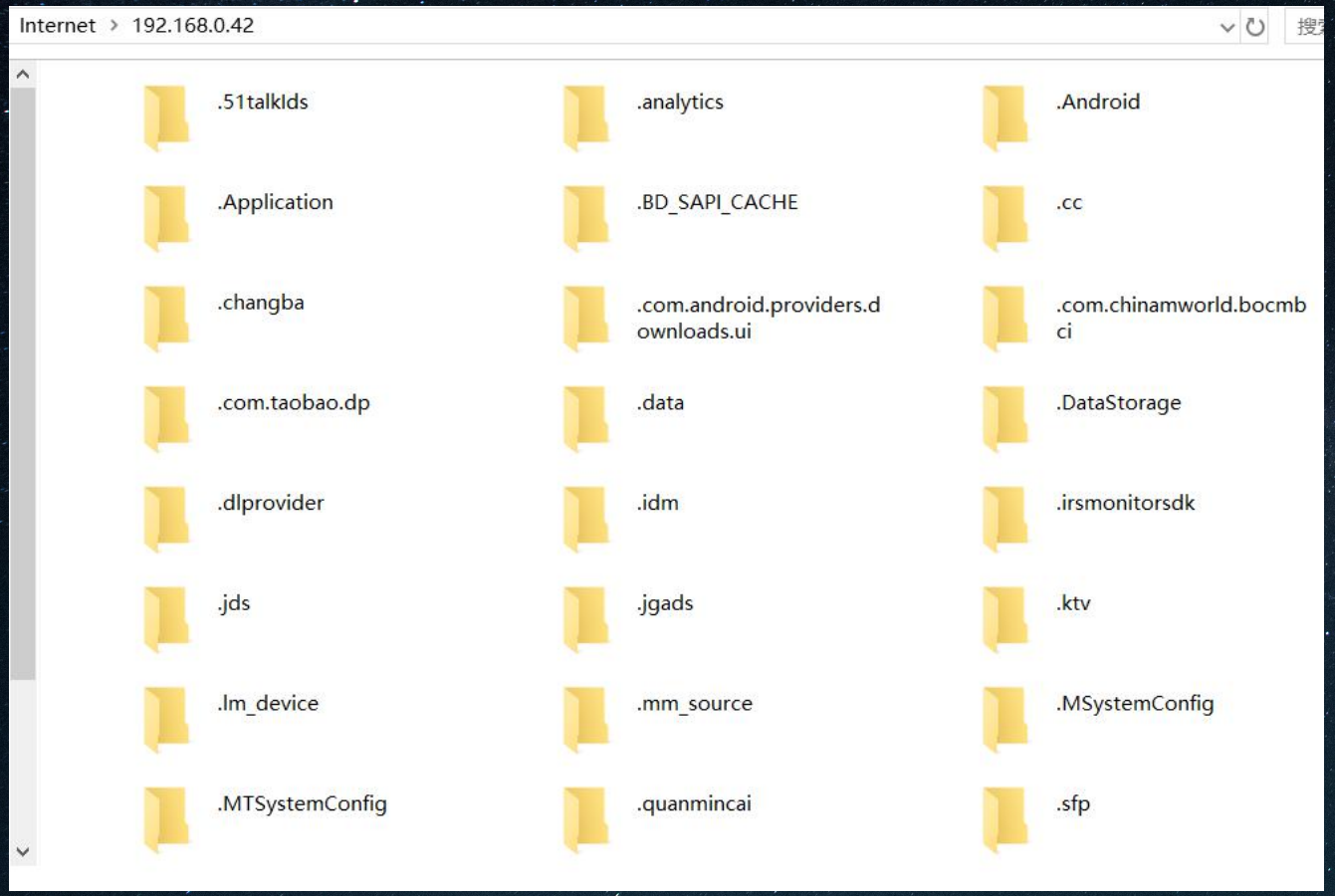
Remote file management on computer

腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB



RFM Vulnerabilities

- Anonymous users have READ and WRITE permissions



Summary



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

- None-confirm before WiFi Connection Established
- Not well protected access to WiFi/WiFi P2P
- Unsecure Transport
- No anti-spoofing

Nearby Sharing Apps Security Survey

Pre-installed Nearby Sharing Apps



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

App	Confirm before WiFi Connection Established	WiFi/WiFi P2P Authentication	Secure Transport	Prevent Spoofing
A	✘	O	✘	✘
B	✘	O	✘	✘
C	✘	O	✘	✘
D	✘	O	✘	✘

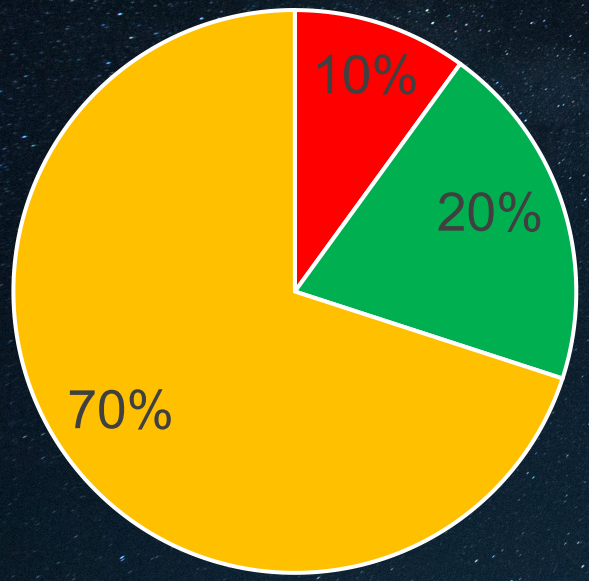
✘ : No security measures
O : Have security measures but can be bypassed
✓ : Have security measures

Top 10 Third Party Nearby Sharing Apps

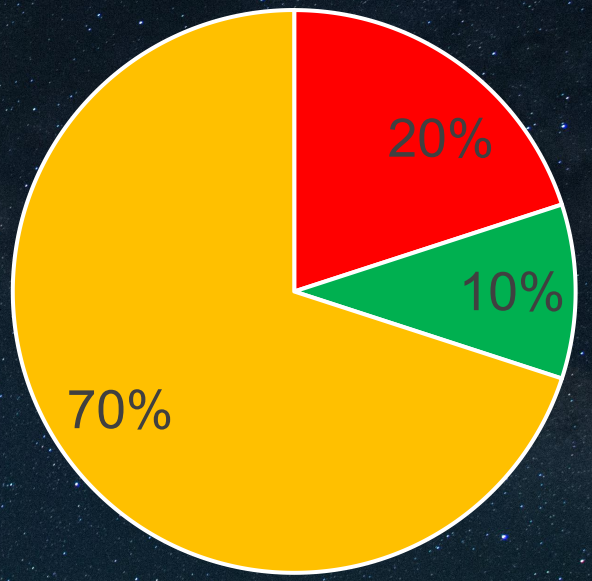


腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

Sniffer Attack



Spoofing



■ : Vulnerable
■ : Safe
■ : Partial Vulnerable

Mitigation

Mitigation



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

- More secure Wi-Fi / Wi-Fi P2P Key Exchange
- Transport encryption
- Prevent spoofing
- Others tips

Secure WiFi/WiFi P2P Key Exchange



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

- Transfer the KEY over a secure channel
  - Out of band: NFC
  - Certificate based key-exchange mechanism
  - Transferring keys over the Internet securely
- PIN / PBC

Secure Transport



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

- Using TLS/HTTPS instead of TCP/HTTP
- Transferring keys over the Internet securely
- Pre-exchanged keys

Prevent spoofing



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

- Certificate based authentication
  - Unique ID
    - Distributed by the server
  - Signed with pre-defined trust anchor
- Attacker can’t sign the Unique ID correctly
- Make sure that the device can distinguish between real users and attackers

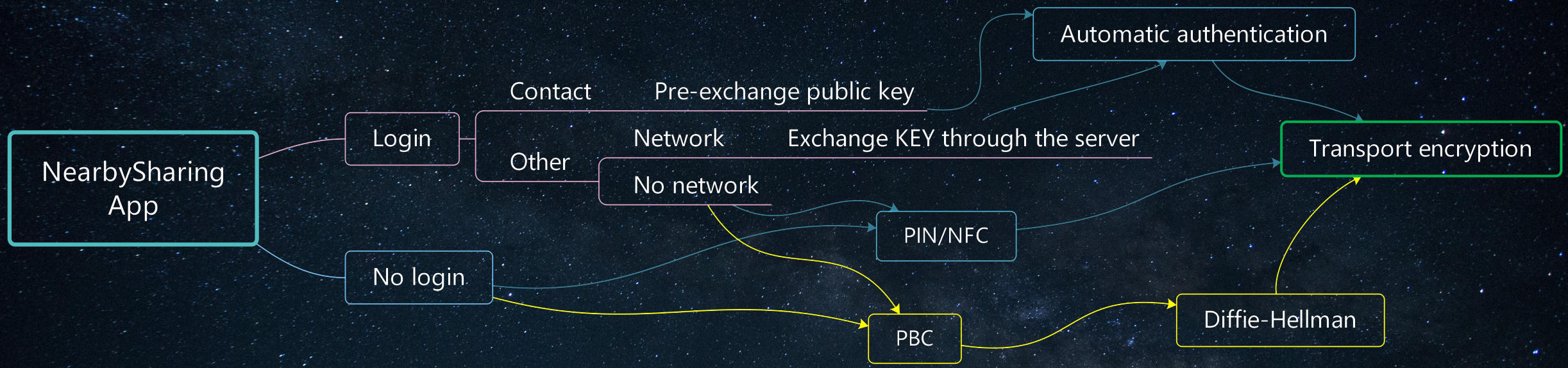
Others tips



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

- Prevent attackers from joining the network
- Turned off by default, turned off after a period of idle time
- Confirm before connection established
- **Do not establish connection automatically with users who are unidentifiable**

Recommended design



Conclusion



腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

- Attack vectors of nearby sharing apps
- First comprehensive research about them
- Found Many vulnerabilities categorized in several common attack methods
- Present some effective mitigation methods

Q&A