
black hat®
ASIA 2017
MARCH 28-31, 2017
MARINA BAY SANDS / SINGAPORE

black hat ASIA 2017

3G/4G Intranet Scanning and its Application on the WormHole Vulnerability

Zhang Qing

Bai Guangdong

Self Introduction

- Zhang Qing
  - Senior Android security researcher from Xiaomi Inc., China
  - Research on Android security and payment security
- Bai Guangdong
  - Lecturer from Singapore Institute of Technology (SIT), Singapore
  - Research on mobile security and protocol analysis
  - Presented “Authenticator leakage in Android” on Black Hat Europe 2015

Agenda

- Introduction and Background
  - 3G/4G intranet
  - Attack surface of 3G/4G intranet
- Scanning 3G/4G intranet
  - Scanner Setup
  - Introduction to WormHole vulnerability
  - Scanning Results and Statistics
  - Countermeasures
- A Honeypot on 3G/4G intranet
  - Findings
- Summary and Take-aways

Introduction and Background

- 3G/4G intranet
- Attack surface of 3G/4G intranet

Cellular Networks: Where are We?

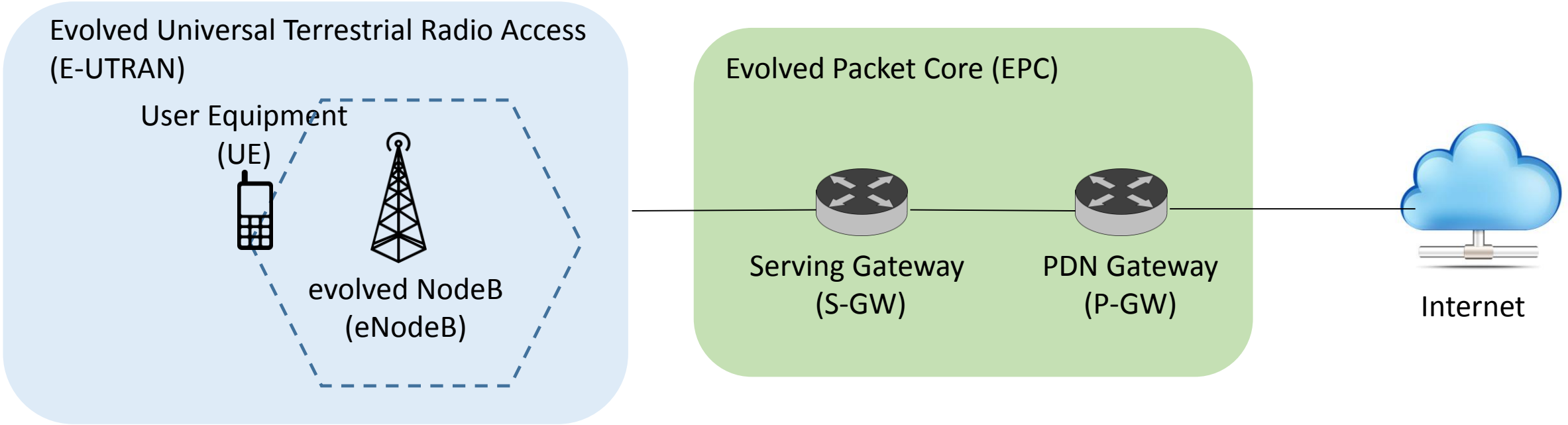
- 1st Generation **Analog** Systems
  - Analog Telecommunication
  - No data transmission, only voice transmission

- 2nd Generation **Digital** Systems
  - Purely digital technology
  - **Circuit switching**: dedicated point-to-point connections during calls
  - TDMA, GSM, CDMA
  - Circuit-switched data services (HSCSD)
  - Very slow data transmission

Cellular Networks: Where are We?

- 2.5 – 3rd Generation
  - Mix of circuit switching and packet-switching
  - Packet-switched data
  - Allows mobile networks to transmit IP packets to the Internet
  - GPRS, EDGE, CDMA2000
- 4th Generation
  - All IP-based secured packet switched network (IPv6 supported)
  - Voice also transmitted over IP
  - LTE, WiMAX

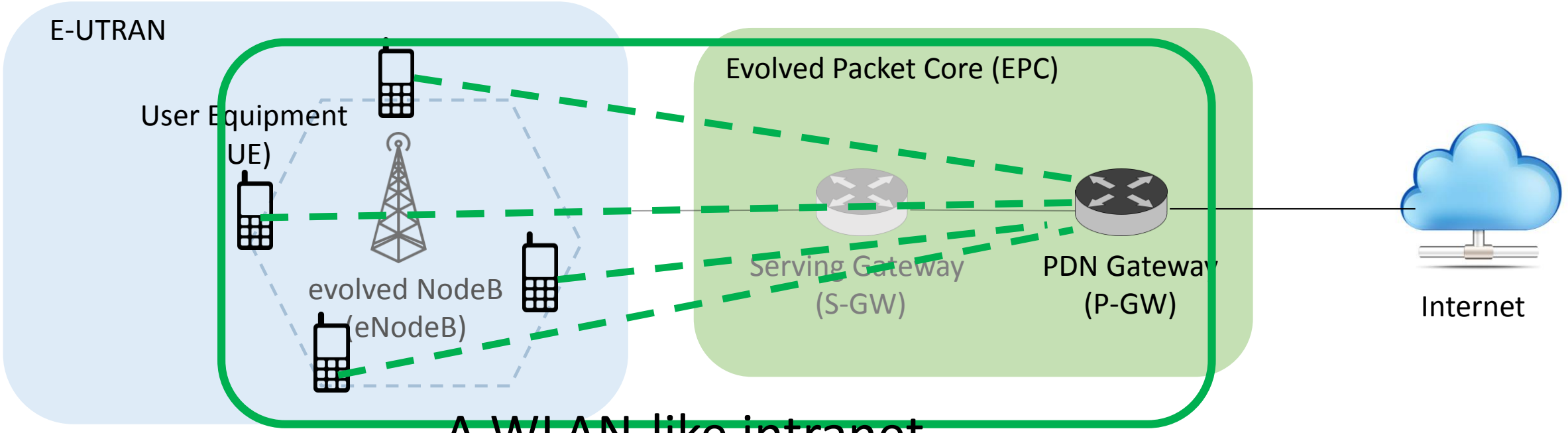
LTE System Architecture



- ✓ E-UTRAN consists of eNodeBs (i.e., base stations).
- ✓ It **manages the radio communication between eNodeB and UE** and **facilitates communication between the UE and EPC**

- ✓ S-GW: All user IP packets are transferred through the S-GW, which serves as the local **mobility anchor** when the UE moves between eNodeBs.
- ✓ P-GW: The PDN (packet data network) Gateway is responsible for **IP address allocation** for the UE, QoS enforcement and flow-based charging.

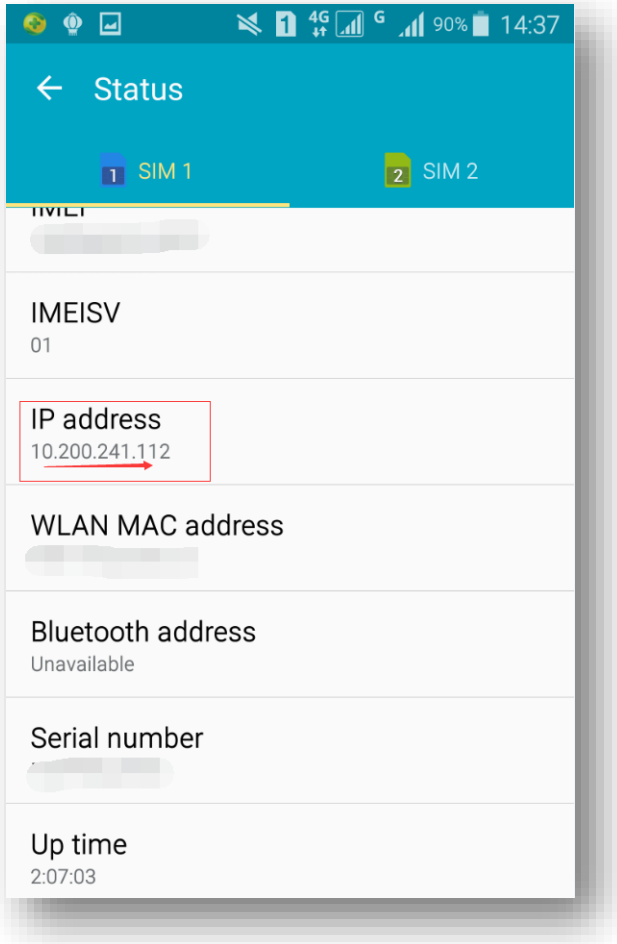
Abstraction of 3G/4G intranet



A WLAN-like intranet

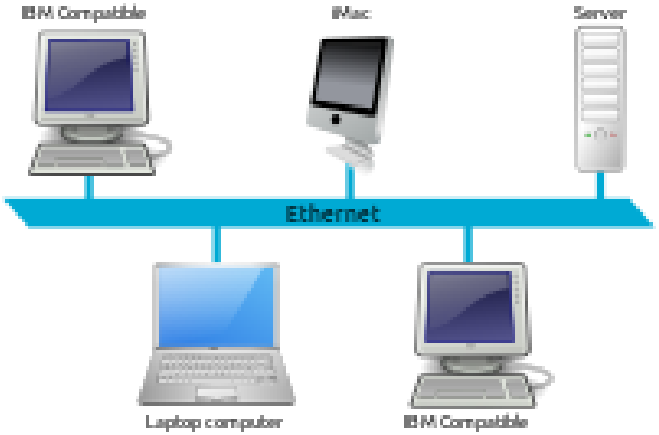
When a UE is connected to 3G/4G network, it is assigned a private IP address.

Example of a Private IP Address



When a mobile phone is connected to 3G/4G network, it is assigned a private IP address within the range of 10.0.0.0 – 10.255.255.255

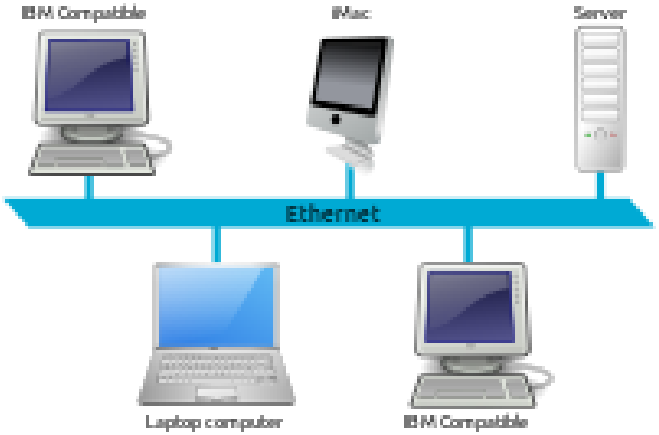
Security in LAN and WLAN



- ✓ Nodes are physically close
- ✓ Used in a limited area such as a residence, laboratory and office, which is relatively **more trustworthy** and **easier to audit**

- ✓ Various security countermeasures, e.g., Wired Equivalent Privacy (WEP) and Wi-Fi Protected Access (WPA)
- ✓ Protected by authentication on APs
  - Difficult for malicious nodes to connect into the intranet

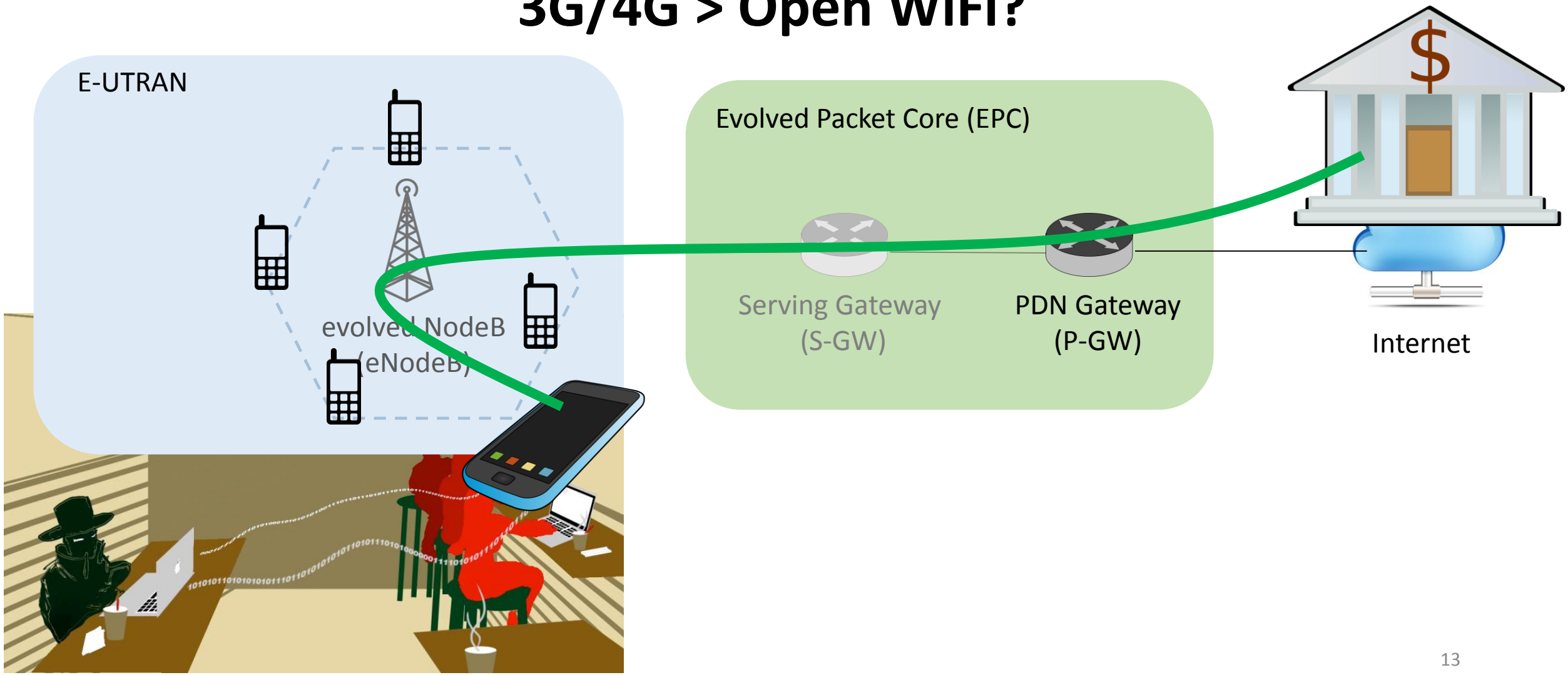
Security in LAN and WLAN



- ✓ Nodes are physically close
- ✓ Used in a limited area such as a residence, laboratory and office, which is relatively **more trustworthy** and **easier to audit**

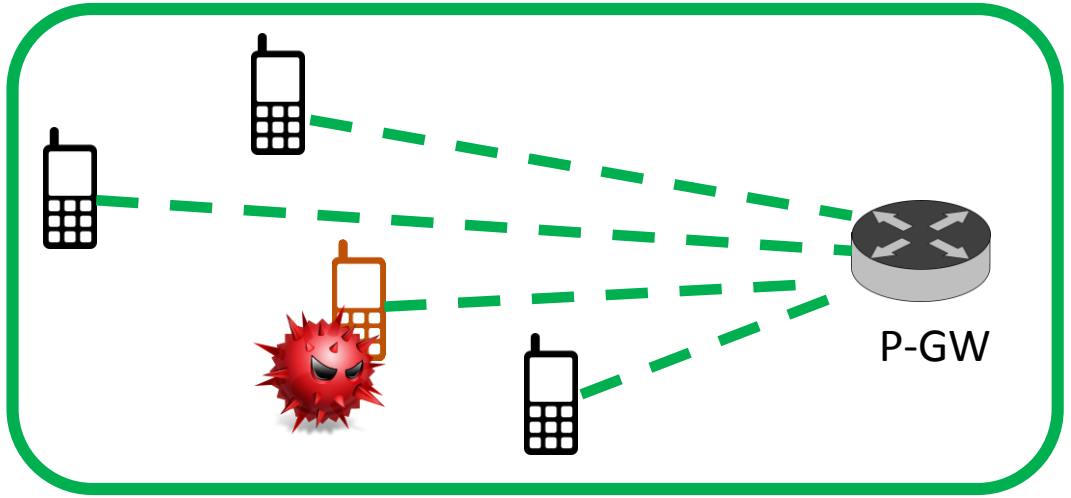
- ✓ Various security countermeasures, e.g., Wired Equivalent Privacy (WEP) and Wi-Fi Protected Access (WPA)
- ✓ Protected by authentication on APs
- ✓ **Insecurity of open WiFi** becomes more and more realized

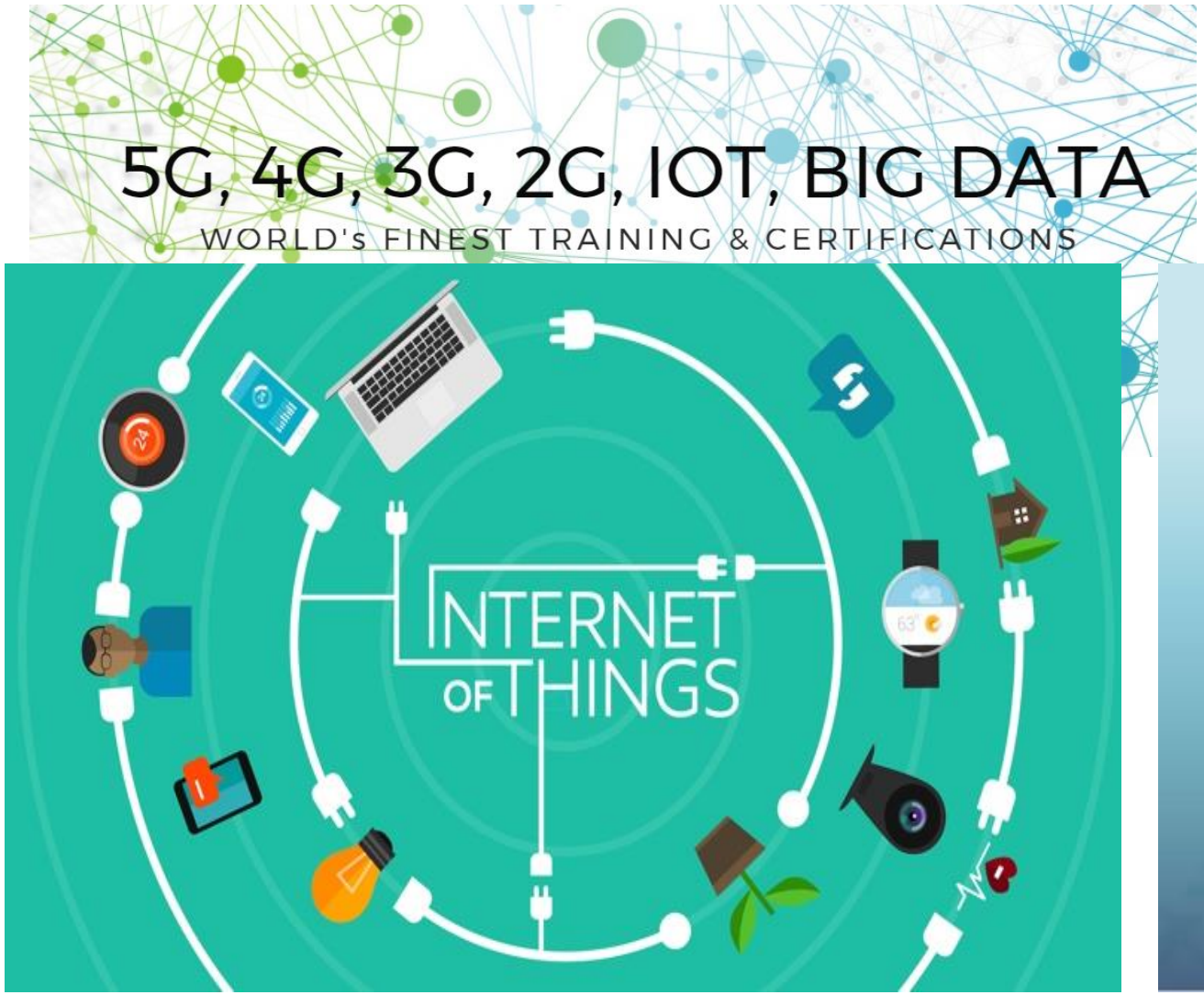
3G/4G > Open WiFi?

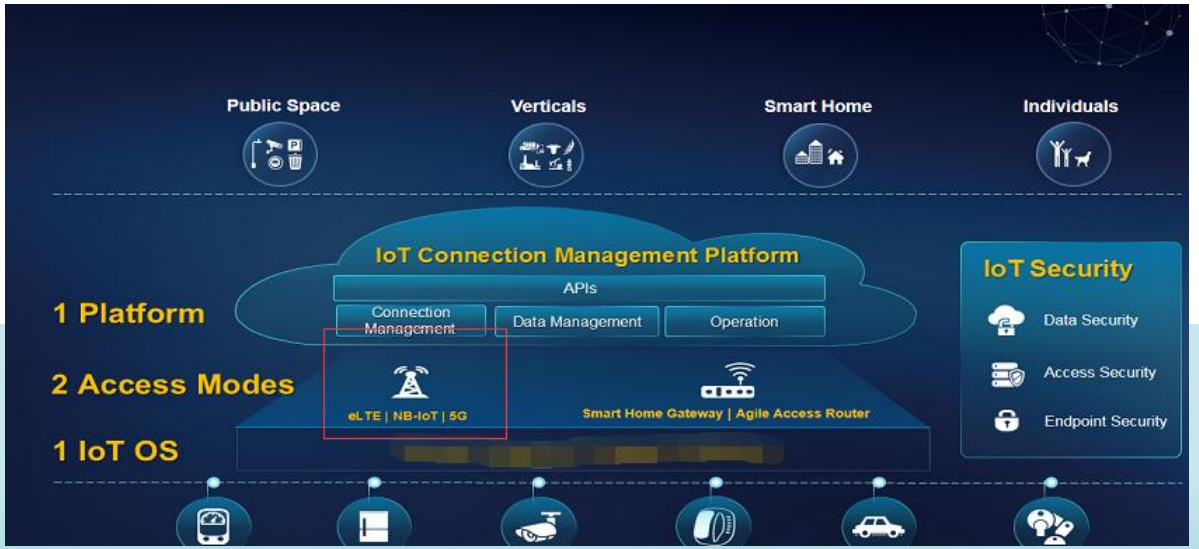


The Dark Forest of 3G/4G Intranet

- A UE has no idea that
  - which **intranet** it is connected in, and
  - its **neighbors** are trustworthy or not
- An 3G/4G intranet is **dynamic**
  - UEs in a intranet are not necessarily connected to the same base station, and vice versa
  - A UE may join and exit dynamically
  - A UE may not be connected to the same intranet each time




5G, 4G, 3G, 2G, IOT, BIG DATA
WORLD's FINEST TRAINING & CERTIFICATIONS
INTERNET OF THINGS

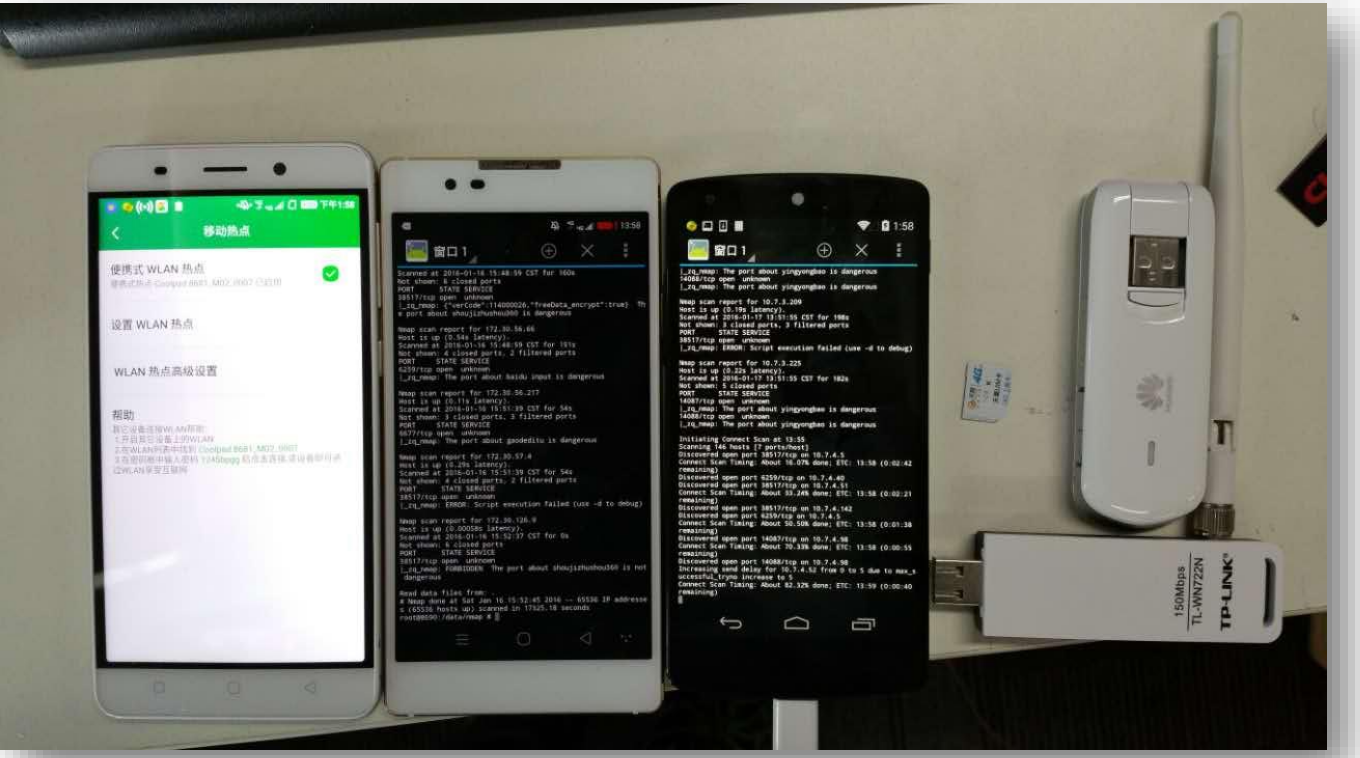

Public Space
Verticals
Smart Home
Individuals
IoT Connection Management Platform
APIs
Connection Management
Data Management
Operation
1 Platform
2 Access Modes
1 IoT OS
eLTE | NB-IoT | 5G
Smart Home Gateway | Agile Access Router
IoT Security
Data Security
Access Security
Endpoint Security


3G/4G
POWERED

Scanning 3G/4G Intranets

- Scanner Setup
- Introduction to WormHole vulnerability
- Scanning Results and Statistics
- Countermeasures

Devices



- 4G Wireless Router
  - which allows us to conduct scanning on a desktop
  - Huawei EC3372-871 4G FDD TD-LTE Cat4 USB Dongle
  - **Scalability**
- A desktop
- 4G Sim Card and Android Smart Phone
  - which allows us to conduct scanning on various places
  - **Mobility**

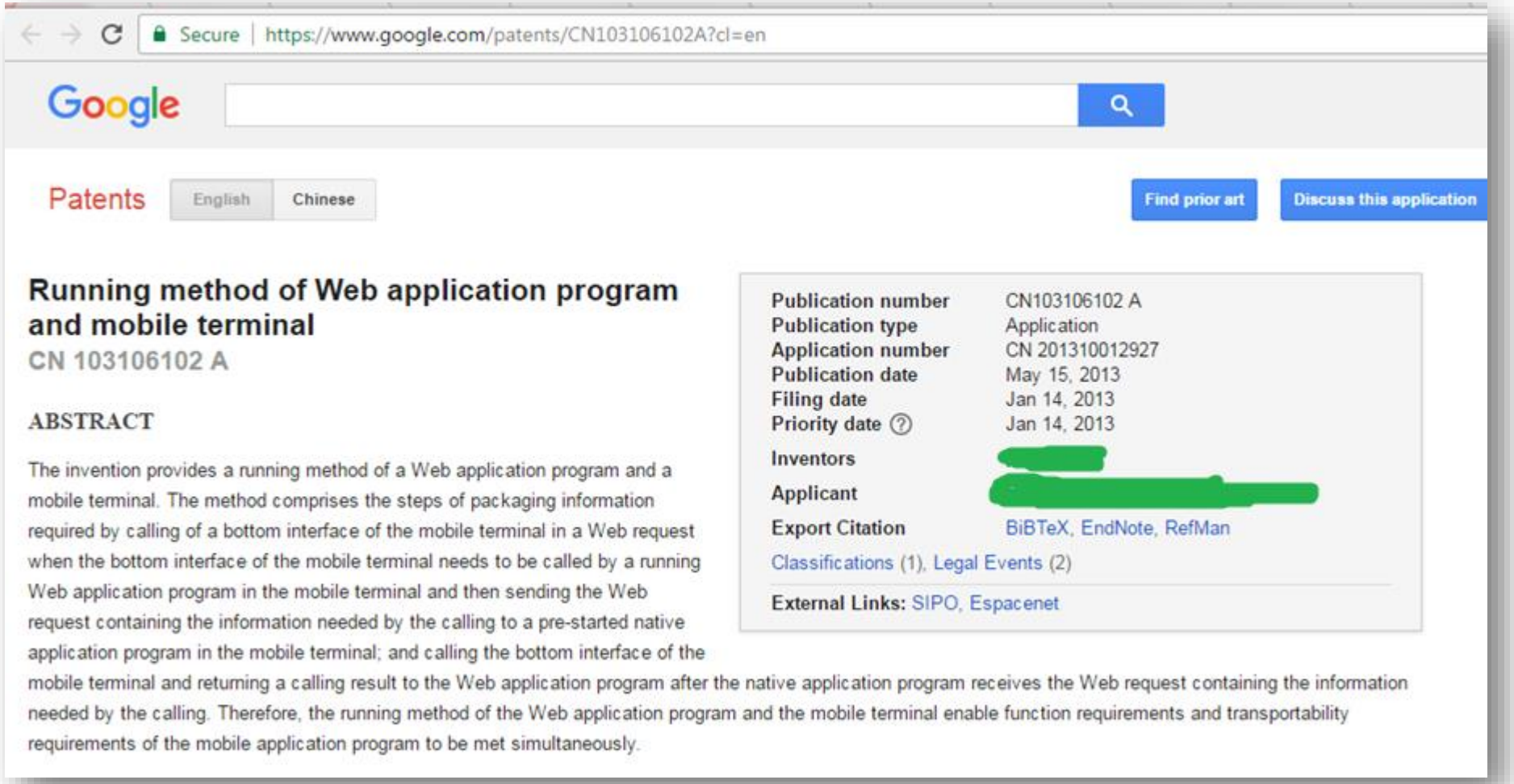
Case Study: WormHole Vulnerabilities

- Android/PUP.WormHole.A [1], [2]
  - Was reported in Oct 21st, 2015
  - Was found in Baidu's SDK Moplus (Port 6592 and 45310)
  - 14,000+ apps got infected [3], 100M users were at risk [4]
- Other vulnerabilities of the same type are found in other major apps
  - 360 Browser (6587, 3851, etc.)
  - Gaode maps (6677)
  - Yingyongbao (14087)

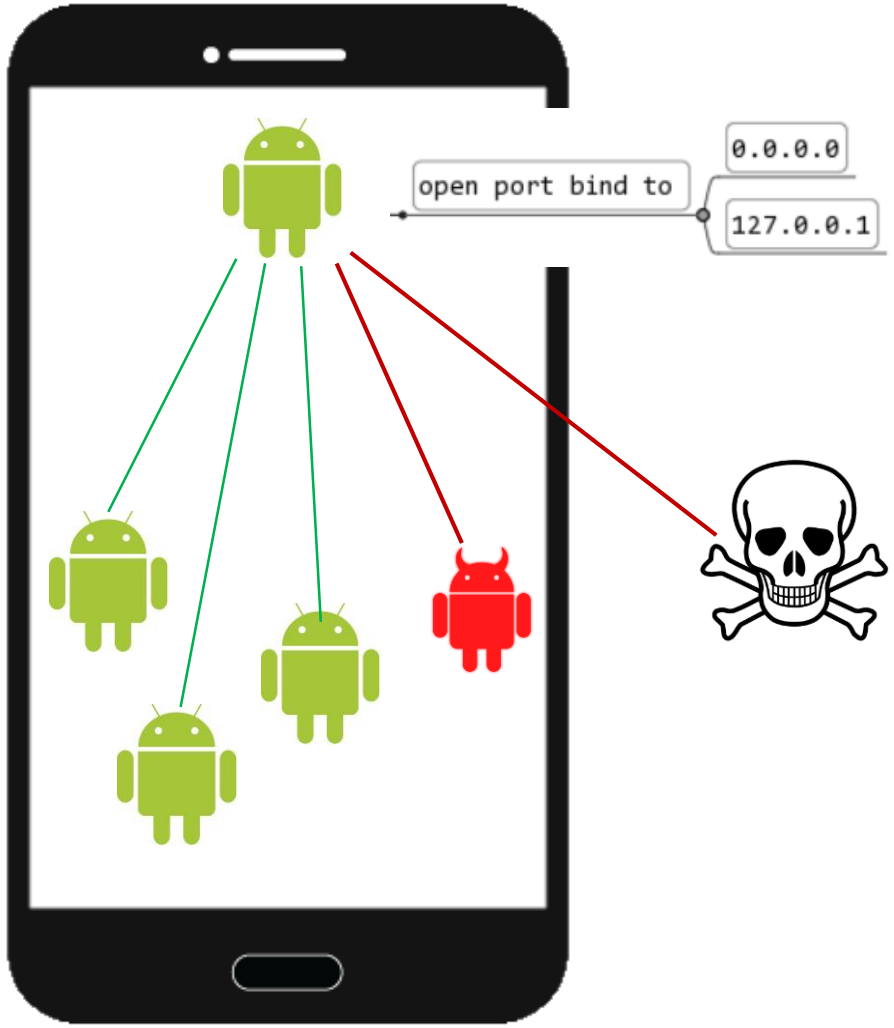
Case Study: WormHole Vulnerabilities

- Why do we target WormHole?
  - This vulnerability is caused by “ImmortalService” – a customized HTTP service used for cross-app communication
  - A proxy acts as a **server**, and opens a **port** for client to invoke it for (**maliciously** or for **functionalities**?)
    - Adding contact information silently
    - Starting any applications by remote control
    - Installing any applications silently
    - Uploading local files to a remote server
    - Getting personal information such as GPS location, IMEI, and an installed applications list

Benign or Malicious?



Benign or Malicious?



Once the proxy opens a port, not only its companions can access it, but also **malicious apps** on the same device and a **network attacker** outside the device can abuse it

Our Approach

- We tested on three telecom operators in China (anonymized below)

Operator A	Operator B	Operator C
10.163.69.0/24	100.112.0.0/16	10.26.0.0/16
10.93.111.0/24	100.101.0.0/16	10.26.0.0/16
10.245.219.0/24	100.101.0.0/16	10.28.0.0/16
10.10.240.0/24	100.119.0.0/16	10.29.0.0/16
	100.119.0.0/16	10.9.0.0/16
	100.101.0.0/16	10.1.0.0/16
	100.97.0.0/16	10.26.0.0/16
	100.114.0.0/16	10.7.0.0/16

  - Different time, different locations
  - We did not test liveness (discuss shortly)

Our Approach

- Tool: nmap
  - ./nmap -sT -p6677,6587,38517,6259,40310,14087,14088 **-T1** -vv **-n -PN** --open **--script test_nmap** -oN lt1026.txt 10.26.0.0/16
  - Challenge: avoid being blocked by firewall and IDS

Parameter	Description
-PN (Treat all hosts as online -- skip host discovery)	Necessary when scanning the Operator B network and multithreading is not suggested. It will be **detected** by IDS if 'PN' is not specified or multithreading is used.
-n	Suggested, Never do DNS resolution
-T 0 or 1 (Set timing template <0-5>, higher is faster)	Have to use this parameter to control the **pace**, in order to avoid the IDS detection when scanning the Operator B network

Script Snippet

```
action = function(host,port)
        if(port.number == 6587) then
            local url = "/t=0"
            local response = http.get(host, port, url)
            if(response==nil or response.body==nil) then
                return "the port about 360 browser is not dangerous "
            end
            local index=string.find(response.body,"\"code\":\"0\"")
            if(index==nil) then
                return  response.body.."the port about 360 browser is not dangerous "
            else
                return  response.body.." and the port about 360 browser is dangerous "
            end
        end
        if(port.number==40310) then

            --accessPage("http://"..host..":40310/sendintent?callback=123&mcmdf=inapp_xxx&intent=inte
            dandroid.intent.action.VIEW%3bend%3b")
            zq, zq2=accessPage("http://"..host["ip"]..":40310/getcuid?callback=123&mcmdf=inapp_xxx")
            return zq2.."    The port about baidu is dangerous"
        end
        if(port.number==6259) then
            zq, zq2=accessPage("http://"..host["ip"]..":6259/getcuid?callback=123&mcmdf=inapp_xxx")
            return zq2.."    The port about baiduinput is dangerous"
```

Different headers and response processing per different ports

Scanning using Android Devices *

- Step 1: push *nmap* to Android's */data/nmap* folder
- Step 2: assign it execution permission using *chmod*
- Step 3: push the script file 'test_nmp.nse' to */data/nmap/scripts*
- Step 4: use nmap under the */data/nmap* folder

*A rooted device required

Scanning Result

	360 Browser	360 Zhushou	Baidu	Baidu IME	Gaode Maps	Yingyongbao
Operator B	61	163	116	253	68	483
Operator C	53	295	161	494	255	539
Operator A	Blocked					

This scanning was conducted on January 25th, 2016

- nmap is blocked by Operator A's firewall strategy. **Alternative** is discussed shortly
- Unfortunately, we cannot estimate infection rate, without knowing the device alive

Sampling of Scanning Speed

Command: nmap -sT -p6868,80,6259,38517,8822,43633 --open -vv $subnet -n -PN

	Subnet	#IP Address	# up Host	Time (Second)
Operator B	100.119.100.0/24	256	1	64.65
	100.119.0.0/16	65,536	33	26,248.47
Operator A	10.93.111.0/24	256	0	310.41
Operator C	10.28.221.0/24	256	5	3,887.76

This scanning was conducted on January 7th, 2016

Alternative Scanner: scapy

- nmap failed to detected any up host on Operator A network
  - May be because of Operator A's firewall
- We use scapy as an alternative after exploration
  - Step 1: we use null scan to detect whether the port is open
  - Step 2: if the port is open, we use *sr()* to send our package and receive the response
- So far we are able to scan ip/24 of Operator A network

Alternative Scanner: scapy

Example of scapy script which probes port 38517/38518 of an IP address

```
stealth_scan_resp = sr1(IP(dst=dst_ip)/TCP(dport=(38517,38518),flags="S"),timeout=10)

print stealth_scan_resp

if(str(type(stealth_scan_resp))=="<type 'NoneType'>"):
    print "Filtered"
elif(stealth_scan_resp.haslayer(TCP)):
    if(stealth_scan_resp.getlayer(TCP).flags == 0x12):
        send_rst = sr(IP(dst=dst_ip)/TCP(dport=(38517,38518),flags="R"),timeout=10)
        print "Open"
    elif (stealth_scan_resp.getlayer(TCP).flags == 0x14):
        print "Close"
elif(stealth_scan_resp.haslayer(ICMP)):
    if(int(stealth_scan_resp.getlayer(ICMP).type)==3 and int(stealth_scan_resp.getlayer(ICMP).code) in [1,2,3,9,10,13]):
        print "Filtered"
~
```

Ethical Consideration

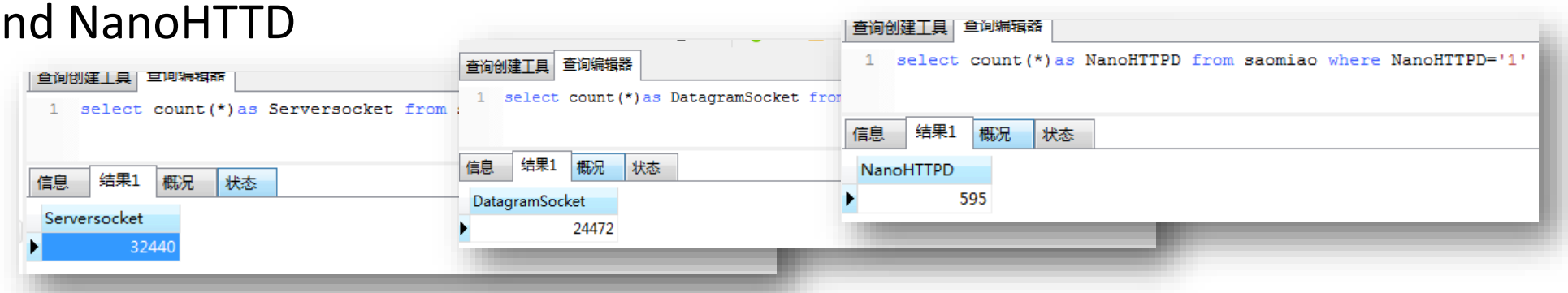
- We collaborate with app developers to notify users for patching if possible
  - We conducted another round of scanning after 3 months
  - Infection number drops significantly

	360 Browser	360 Zhushou	Baidu	Baidu IME	Gaode Maps	Yingyongbao
Operator B	1 / 61	7 / 163	29 / 116	9 / 253	2 / 68	82 / 483
Operator C	17 / 53	55 / 295	54 / 161	154 / 494	73 / 255	189 / 539
Operator A	Skipped					

num after / is number of infection 3 months ago
This scanning was conducted on April 22th, 2016

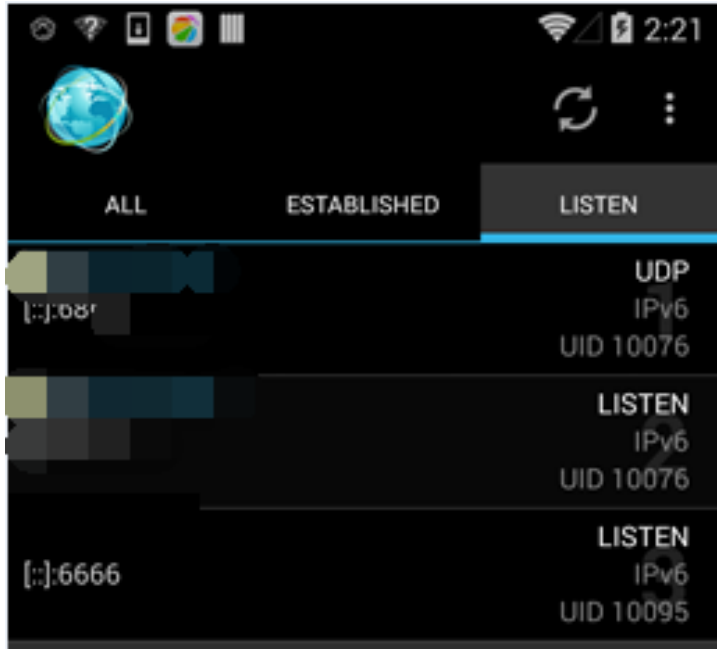
Ethical Consideration

- We collaborate with app developers to notify users for patching
  - We conducted another round of scanning after 3 months
  - Infection number drops significantly
- Vetting apps in the market (ongoing)
  - We have crawled 200,000 apps from an app market in China
  - We use a pattern matching to find apps using ServerSocket, DatagramScoket and NanoHTTD



A Case Study

- A popular app in China, which has 11M installs
  - Anonymized for security of the users



- Open a *ServerSocket* and listen on port 6666
- Receive commands from any other clients
  - JUMPTO_activity Jump to an activity
  - VERSION Version number
  - INFORMATION info of the phone
  - *****(anonymized) Start its normal functionality
  - CANCEL***** Stop its normal functionality
  - … …

A Case Study

- In its newer version, it uses BASE64 to encode the commands and an "encryption" which XOR a number generated from a random seed N
  - $cipher = base64\_encode(command) \oplus F(seed)$
  - But, $F()$ is not important at all
  - $command = base64\_decode(\ cipher \oplus F(seed)\ )$
  - How should we get seed?
- Seed is generated by client and sent to server once client receives a command-in-plain-text "*versionex*"
- Security by obscurity: no security at all

A Honeypot on 3G/4G Intranets

- Honeypot Setup
- Findings

Honeypot Setup

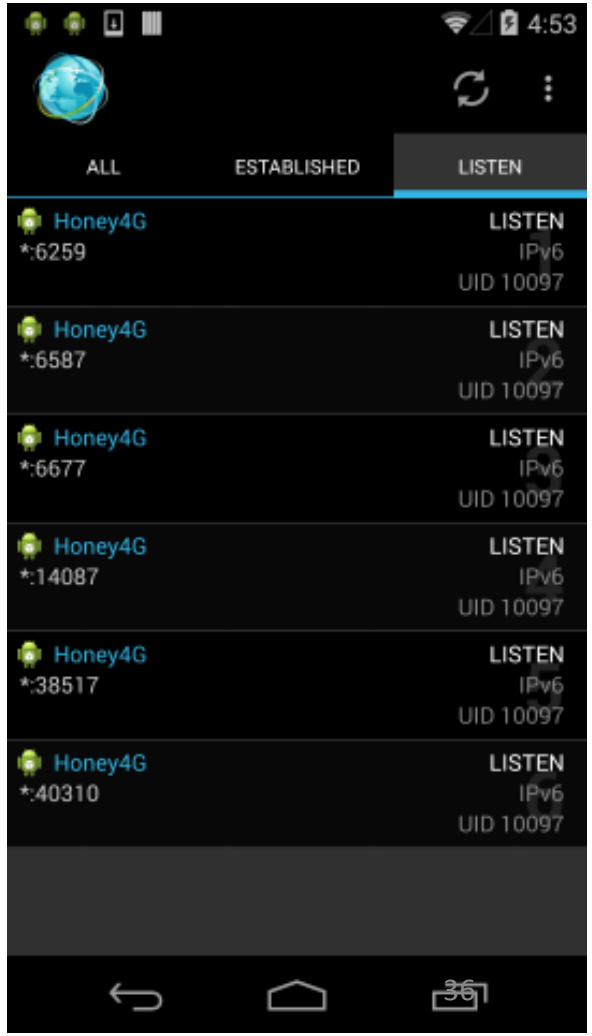
- 4 Honeypots over 4 Cities
- 4G Wireless Router & Desktop
  - Huawei EC3372-871 4G FDD TD-LTE Cat4 USB Dongle
- Modern Honey Network[1]
  - A free open source software which supports honeypot deployments

full ports mapping

Customized Pot on Mobile Device

```
public int onStartCommand(Intent intent, int flags, int startId) {
    // TODO Auto-generated method stub
    ChatRoomServer server = new ChatRoomServer(6677);  //gaode
    server.startServer();
    ChatRoomServer server1 = new ChatRoomServer(6259);  //BaiDu input
    server1.startServer();
    ChatRoomServer server2 = new ChatRoomServer(40310);  //baidu browser
    server2.startServer();
    ChatRoomServer server3 = new ChatRoomServer(14087);  //tencent yingyongbao
    server3.startServer();
    ChatRoomServer server4 = new ChatRoomServer(6587);  //360 browser
    server4.startServer();
    ChatRoomServer server5 = new ChatRoomServer(38517);  //360 apps market
    server5.startServer();
    //return START_STICKY;
    //return super.onStartCommand(intent, flags, startId);
    Notification notification = new Notification();
    startForeground(1, notification);
    return START_STICKY;
}
```

- 6 known WormHole ports
- Feed information the attacker needs, while recording the attacks



Results

- Each honey pot is scanned once a day on average
  - 3G/4G intranet scanning **has been used**
  - Known WormHole vulnerabilities have been extensively exploited

- Trace attackers

```
/100.101.118.55 47736 GET /77 HTTP/1.1 on 6259 01/14 14:20:22
/100.101.118.55 47736 Host: 100.114.0.124:6259 on 6259 01/14 14:20:22
/100.101.118.55 47736 Connection: keep-alive on 6259 01/14 14:20:22
/100.101.118.55 47736 Cache-Control: max-age=0 on 6259 01/14 14:20:22
/100.101.118.55 47736 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/
/100.101.118.55 47736 x-wap-profile: http://218.[redacted]MTK_LTE_Phone_L_UAprofil
/100.101.118.55 47736 User-Agent: Mozilla/5.0 (Linux; U; Android5.0; zh-cn; YL-Coolpad 868
/100.101.118.55 47736 Accept-Encoding: gzip, deflate on 6259 01/14 14:20:22
/100.101.118.55 47736 Accept-Language: zh-CN,en-US;q=0.8 on 6259 01/14 14:20:22
/100.101.118.55 47736 X-Requested-With: com.android.browser on 6259 01/14 14:20:22
/100.101.118.55 47736  on 6259 01/14 14:20:22
/100.114.0.124 50646 GET /getClientInfo HTTP/1.1 on 38517 01/15 17:03:41
/100.114.0.124 57802 GET /t=0 HTTP/1.1 on 6587 01/15 17:03:41
```

http://218.*.*.*/*/PhoneModel

Trace attackers

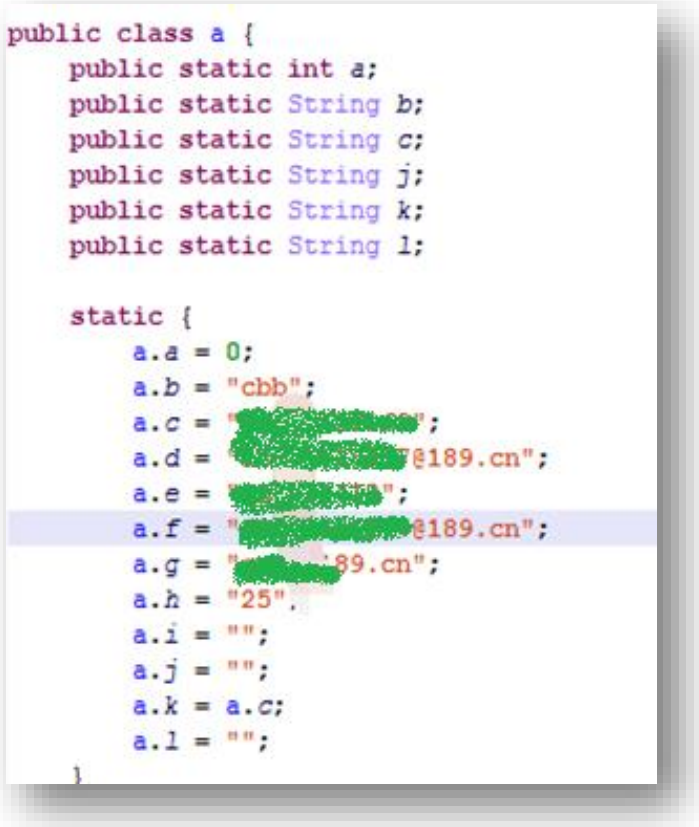
http://218.*.*.*/*/PhoneModel

GitHub

- Under an open source project which **offers anonymous web access**, the owner of this IP asked a question why this *x-wap-profile* is added, in **2014**
  - He/she was doing scanning in an anonymous way
  - He knew WormHole in **2014**?

An Attack Detected by our Honey Pot

- Someone uses a WormHole to install a spyware located in http://ada**dh.com/qr.apk
- The spyware reads the SMS messages and sends them to an email address
- The email address and password are found in the apk file



```
public class a {
    public static int a;
    public static String b;
    public static String c;
    public static String j;
    public static String k;
    public static String l;

    static {
        a.a = 0;
        a.b = "cbb";
        a.c = "[redacted]";
        a.d = "[redacted]@189.cn";
        a.e = "[redacted]";
        a.f = "[redacted]@189.cn";
        a.g = "[redacted]89.cn";
        a.h = "25";
        a.i = "";
        a.j = "";
        a.k = a.c;
        a.l = "";
    }
```

Summary and Take-aways

Summary & Take-aways

- 3G/4G intranet is **more open** and **dynamic** than LAN and WLAN
  - It is possible to conduct a large-scale scan over the 3G/4G intranet
  - It may have been exploited earlier
  - It is a potential risk for IoT devices
- Be ware of this attack surface
  - Operators: Intrusion Detection System required
  - App developers: authentication is necessary if open any service
    - More research on security of sockets in Android apps [CCS'16]

[CCS'16] Y. Shao, J. Ott, Y. J. Jia, Z. Qian, and Z. M. Mao. The Misuse of Android Unix Domain Sockets and Security Implications.

Questions?

Zhang Qing
Bai Guangdong