New Ways of IPV6 Scanning

Shupeng Gao, Xingru Wu, Jie Gao
Baidu Security Lab
• Introduction of the IPV6
• The risks and new scanning methods
• How to exploit
• Suggestions and summary
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?

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

Scanning methods discovered by security researchers:
Traverse low bit address: for example, 2401:0a0b::0~ 2401:0a0b::fff
Generate IPV6 address according to MAC address
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
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
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)  Pixel 4 connect 4G LTE
- ICMPv6 Option (Recursive DNS Server 240e:40:8000::10)
```

In home broadband

- GPON device will obtain a /64 prefix from operator, generate its WAN addr
- Then use WAN addr and DHCPv6, to get a 64 prefix as its LAN addr
- or a /60 prefix as LAN addr, for the lower layer router to continue to allocate 64 bit prefix
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
• Introduction of the IPV6
• The risks and new scanning methods
• How to exploit
• Suggestions and summary
The demonstrations of obtaining the full address by sending special ICMP packets.
<table>
<thead>
<tr>
<th>Risk</th>
<th>Scan worldwide</th>
<th>Android</th>
<th>IOS</th>
<th>Linux</th>
<th>Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Risk 1</strong></td>
<td>ICMP unreadable error return the full addr</td>
<td>Y</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Hotspot</td>
</tr>
<tr>
<td><strong>Risk 2</strong></td>
<td>In some cases the IPv6 addr will become shorter</td>
<td>Y</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Risk 3</strong></td>
<td>IPv6 addr can be sniffed and calculated from radio nearby</td>
<td>N</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Risk 4</strong></td>
<td>ICMP time exceeded error returned the full addr (all Linux kernel based devices)</td>
<td>Y</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Hotspot or Forward</td>
</tr>
<tr>
<td><strong>Risk 5</strong></td>
<td>All zero address returned the full addr (all Linux kernel based devices)</td>
<td>Y</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Hotspot or Forward</td>
</tr>
</tbody>
</table>

**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.
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
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
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
- ChangPing district
- Random
- Beijing and 4/5G net
<table>
<thead>
<tr>
<th>District name</th>
<th>China Mobile broadband</th>
<th>China Mobile 4/5G</th>
<th>China Unicom broadband</th>
<th>China Unicom 4/5G</th>
<th>China Telecom broadband</th>
<th>China Telecom 4/5G</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dongcheng, Beijing,</td>
<td>2409:8a00::-24</td>
<td>2409:8900::-24</td>
<td>2408:8206::-24</td>
<td>2408:8406::-24</td>
<td>240e:304::-24</td>
<td>240e:404::-24</td>
</tr>
<tr>
<td>China</td>
<td>09:8a00:bf::</td>
<td>09:8900:bf::</td>
<td>08:8206:bf::</td>
<td>08:8406:bf::</td>
<td>e:304:bf::</td>
<td>e:404:bf::</td>
</tr>
<tr>
<td><strong>110102- Xicheng</strong></td>
<td>2409:8a00::c00:</td>
<td>2409:8900::c00:</td>
<td>2408:8206::c00:</td>
<td>2408:8406::c00:</td>
<td>240e:304::c00:</td>
<td>240e:404::c00:</td>
</tr>
<tr>
<td><strong>District, Beijing,</strong></td>
<td><strong>-2409:8a00::17ff:</strong></td>
<td><strong>-2409:8900::17ff:</strong></td>
<td><strong>-2408:8206::17ff:</strong></td>
<td><strong>-2408:8406::17ff:</strong></td>
<td><strong>-240e:304::17ff:</strong></td>
<td><strong>-240e:404::17ff:</strong></td>
</tr>
<tr>
<td><strong>China</strong></td>
<td><strong>::</strong></td>
<td><strong>::</strong></td>
<td><strong>::</strong></td>
<td><strong>::</strong></td>
<td><strong>::</strong></td>
<td><strong>::</strong></td>
</tr>
<tr>
<td><strong>110105- Chaoyang</strong></td>
<td>2409:8a00::180</td>
<td>2409:8900::180</td>
<td>2408:8206:180</td>
<td>2408:8406:180</td>
<td>240e:304:1800:</td>
<td>240e:404:1800:</td>
</tr>
<tr>
<td><strong>District, Beijing,</strong></td>
<td><strong>0::-2409:8a00::2:</strong></td>
<td><strong>0::-2409:8900::2:</strong></td>
<td><strong>0::-2408:8206::2:</strong></td>
<td><strong>0::-2408:8406::2:</strong></td>
<td><strong>-240e:304:23ff:</strong></td>
<td><strong>-240e:404:23ff:</strong></td>
</tr>
<tr>
<td><strong>China</strong></td>
<td><strong>3ff::</strong></td>
<td><strong>3ff::</strong></td>
<td><strong>3ff::</strong></td>
<td><strong>3ff::</strong></td>
<td>:</td>
<td>:</td>
</tr>
</tbody>
</table>
Use existing tool, do not return ICMP replay, so do not display.
We can use tshark and tcpdump to monitor, and get returned packets.

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 use
- Build your own web server to collect Information
- 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
Video
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.
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
When:
IPV6 forwarding or hotspot func is enabled

The request prefix is correct

Control the TTL value of ICMP or IP becomes 0

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
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

Now ping 240e:404:7901:1786::0, with full zero addr, will return the ICMP replay pkt, with the full addr
Tested on Surface Pro LTE Advanced
ICMP echo request return the full addr

return the full addr
• Introduction of the IPV6
• The risks and new scanning methods
• **How to exploit**
• Suggestions and summary
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)
<table>
<thead>
<tr>
<th>Affected system</th>
<th>Android</th>
<th>IOS</th>
<th>Windows</th>
<th>Linux Desktop</th>
<th>Embedded Linux (Network access)</th>
<th>Embedded Linux (IoT device)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affected</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>Most</td>
<td>Some</td>
</tr>
<tr>
<td>System</td>
<td>Android 11</td>
<td>IOS 14</td>
<td>Win 10</td>
<td>Ubuntu 20.04</td>
<td>OpenWrt Embedded Linux</td>
<td>Embedded Linux</td>
</tr>
<tr>
<td>Internet access type</td>
<td>LTE / 5G</td>
<td>LTE / 5G</td>
<td>LTE / 5G</td>
<td>LTE / 5G</td>
<td>Home Broadband</td>
<td>LTE / 5G</td>
</tr>
<tr>
<td>Device</td>
<td>Pixel 4</td>
<td>All iPhone</td>
<td>Surface Go / Pro LTE Advanced ThinkPad X1 Carbon 4G LTE LTE USB dongle</td>
<td>ASUS router with PPPoE ZTE GPON ONU</td>
<td>4G LTE Router 4G Pocket Hotspot 5G CPE Samsung Watch with e-sim</td>
<td></td>
</tr>
<tr>
<td>Additional</td>
<td>N</td>
<td>Hotspot enabled</td>
<td>Hotspot enabled</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Amount</td>
<td>Very large</td>
<td>A little</td>
<td>A bit</td>
<td>large</td>
<td>A little</td>
<td></td>
</tr>
</tbody>
</table>

* * * * * A little

* * * * A little
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
Remote control vending machine

video
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

Port-Based vulnerability:

- Operating system vulnerabilities
- APP / service vulnerabilities
- Manager page/tools, ssh / telnet / adb / admin web

DDos attack based on ICMP / UDP

Just for security researching
Do not attack!
Sending all zero address will return full address both on Android and GPON devices

Get millions of addresses in 10 minutes

- All Android
- Part of Linux IoT
- All GPON
- Part of router

**LTE / 5G**

**Home Broadband**

2abc:ddd:eeee:ffff::0
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 its IPV6 address, which opens the port 2018

Use our address scanning

Then send the exp

Already fixed
video
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
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
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

Researchers can analyze, fuzz other kinds of ICMP packets to see if they can be spoofed
• Introduction of the IPV6
• The risks and new scanning methods
• How to exploit
• Suggestions and summary
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
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