KEYSIGHT

What is Dead May Never Die
The Immortality of SDK Bugs

Ricky “HeadlessZeke” Lawshae
IoT Village
DEF CON 33

Introductions

Who am I?

- Principal Security Researcher for Keysight Technologies
- Focus mainly on offensive research of IoT Devices
- Found vulnerabilities in devices from Meta, Mazda, Crestron, HID Global, and many more
- Spoke at conferences around the world including Ruxcon, Recon, Insomnihack, etc
  - 7x DEF CON speaker (5x DEF CON proper, 2x IoT Village)

SDK’s

What are they?

- “Software Development Kit”
  - But really “System Development Kit” in this context
- Collection of applications, drivers, scripts, toolchains, code examples and other artifacts
  - Anything needed to get a chipset integrated into a third party’s ecosystem
- Can include standard and non-standard things
  - Standard things used for boilerplate elements like SSH and web servers
  - Non-standard for proprietary network protocols, etc
- This talk focuses on network chipsets for obvious reasons

SDK’s

Why are they?

- Chipsets are complicated and getting more so everyday
  - Added functionality for increased demand/expectations
  - What was once a single-purpose IC or MC now may be a full SoC
- Chipset manufactures want maximum adoption
  - Need to make it quick and easy for a third-party vendor to integrate
  - Minimize vendor dev time/effort
  - Keep costs down
- Chipsets change ownership...a lot…
  - Companies merge or get bought out
  - Inherited lines can still be supported/sold without starting from square one

SDK’s

What could go wrong?

- Leads to unintended behaviors
  - Extra services running without vendor knowledge
  - Unpolished example code used in production
- Out-of-date versions
  - Every element in an SDK needs to keep up with patches
  - Some OSS projects get forked/modified and never pull in updates from main
- Scattered install base hinders patch dissemination
  - A vuln in a single chipset can affect full product lines across many different vendors
  - Some vendor products EOL quickly
  - Impossible to ensure every product gets patched even if a vendor is aware

SDK’s

Also Note...

- Not every chip maker uses the same SDK for their entire product line
  - But there will be commonalities, especially in proprietary services
  - Source code can be hard to find without an NDA/dev license
- Command injection bugs are best for situations like this
  - Find a single cmd inj bug in an SDK, and exploit works across entire line
  - No need to worry about mem layout or even arch
  - Preferred by botnet authors for the same reason
  - Anyways, CMDINJ4LYFE

Common Chip-Makers and Their Attack Surfaces

Common Chip-Makers and Their Attack Surfaces

Broadcom/Brocade/Cypress/Infineon

- Avago buys Infineon in 2008
- Avago buys Broadcom in 2015, Avago changes its name to Broadcom
- Cypress buys Broadcom’s IoT portfolio (includes WICED SDK) in 2016
- Broadcom buys Brocade in 2017
  - Tries to buy Qualcomm in 2017 too but is denied
- One of the most widely used chipsets
  - Networking, storage (Brocade), IoT (Cypress)
  - Multiple architectures
  - BCM5325E seems to be most common, followed by BCM5357x and BCM53125

Common Chip-Makers and Their Attack Surfaces

Broadcom/Brocade/Cypress/Infineon

- Uses Frankenstein mashup of micro_httpd, mini_httpd, and GoAhead for http
- UPnP implementation is super buggy and possibly ripped off of old Ralink code?
  - https://www.onekey.com/resource/broadcom-sdk-vulnerabilities-bug-reports
- Source code:
  - https://svn.dd-wrt.com/browser/src?rev=45725&order=name

Common Chip-Makers and Their Attack Surfaces

Broadcom/Brocade/Cypress/Infineon

- WICED SDK
  - Wireless Internet Connectivity for Embedded Devices
  - Very old holdover from Infineon
  - Would be surprised if there weren’t devices still using it
  - And here’s some source code:
    - https://github.com/wwwlllllll/bcm-wiced-sdk

Common Chip-Makers and Their Attack Surfaces

Mediatek/Ralink/Inprocomm

- MediaTek buys Inprocomm in 2005
- MediaTek buys Ralink in 2011
- Also extremely common
  - Largest vendor of smartphone chipsets in the world
  - Mostly MIPS (in routers at least) but also ARM
  - MT7620A / MT7620N seems the more widely used in routers
- Have been taking an absolute beating lately
  - 35+ Critical/High CVE’s so far this year
  - CVE-2025-20693 alone affects 22 chipsets

Common Chip-Makers and Their Attack Surfaces

Mediatek/Ralink

- MTK APSoC SDK
  - GoAhead and lighttpd for http
  - Dropbear for ssh
  - miniupnpd, linux-igd, and wsc_upnp for upnp
  - ATED
    - Proprietary diagnostic and QA program
  - REG
    - Proprietary register read/write program
- Source code: https://github.com/dt99/mt7615/tree/master/mt_wifi

Common Chip-Makers and Their Attack Surfaces

Realtek

- No notable acquisitions besides buying Avance in the 90’s
- RTL8186 and RTL8196C are extremely popular, I see them a lot
  - Mix of ARM, MIPS, and Lexra (?)
  - Used by Belkin, D-Link, Netgear, Trendnet, Zyxel, etc
  - Mainly networking...everywhere…
- Seeing more CVE’s start to pop up lately
  - https://www.onekey.com/resource/advisory-multiple-issues-realtek-sdk-iot-supply-chain
  - https://blog.talosintelligence.com/vulnerability-roundup-july-10-2024/

Common Chip-Makers and Their Attack Surfaces

Realtek

- Jungle SDK
  - boa for HTTP
  - wscd for UPnP
  - Proprietary tools
    - MP for diagnostics
    - RRCP – really fun ethernet protocol if you run across it (see my defcon31 talk)

Common Chip-Makers and Their Attack Surfaces

Microchip/Atmel/Micrel

- Microchip buys Micrel in 2015
- Microchip buys Atmel in 2016
- Used in literally everything
- Mostly included to mention ASF...

Common Chip-Makers and Their Attack Surfaces

Microchip/Atmel/Micrel

- Advanced Software Framework
  - https://www.microchip.com/en-us/tools-resources/develop/libraries/advanced-software-framework
  - Open-source code library and docs for everything Atmel
  - 500-page API reference pdf
  - Basically, if it has an Atmel chip, it probably uses this codebase
- And vulns are coming
  - https://thehackernews.com/2024/09/critical-flaw-in-microchip-asf-exposes.html
  - “There exists a vulnerability in all publicly available examples of the ASF codebase…”

Old-Days

Old-Days

CVE-2014-8361 – the reason for this talk

- Cmd inj vuln in Realtek's miniigd SOAP service
  - NewInternalClient field of AddPortMapping command
  - Talked about it at DEF CON 23
  - Released as 0-day after no response from Realtek
- Miniigd service shipped with all rtl81xx based routers
  - 100's of different models from D-Link, TP-Link, Netgear, Linksys, Trendnet, etc
- Fun fact: evidence of being forked from miniupnpd
  - Banner still said miniupnpd in some places
  - Same vuln existed in older version of miniupnpd and had been patched

Old-Days

CVE-2015-1187 – Same story, different vuln

```
<s:Body>

  <u:AddPortMapping xmlns:u="urn:schemas-upnp-
  org:service:WANIPConnection:1">

    <NewInternalClient>`[injected command]`</NewInternalClient>

  </u:AddPortMapping>

</s:Body>
```

Old-Days

CVE-2014-8361 – the reason for this talk

- Realtek eventually dropped miniigd from SDK...too late
- Got picked up by multiple IoT botnets including Mirai
- CVE-2014-8361 recently added to CISA KEV list nearly a decade later
- Still actively exploited
- Still almost 9k devices on shodan with banners indicating vuln versions
- Never going away

Old-Days

CVE-2015-1187 – Same story, different vuln

- Cmd inj in ping.ccp tool of the ncc/ncc2 service
  - Discovered independently by both Peter Adkins (2015) and Tiago Caetano Henriques (2014)
  - Added to CISA KEV in 03/2022
  - Observed in Mirai variants in 2023
- Multiple D-Link and TrendNET devices listed as vulnerable
  - Some Mediatek chipsets, some Ralink, some Realtek
- Searching for ncc/ncc2 mostly leads to other vuln disclosures

Old-Days

CVE-2015-1187 – Same story, different vuln

```
POST /ping.cpp HTTP/1.1
User-Agent: curl/7.54.0
Content-Type: application/x-www-form-urlencoded
Content-Length: 83


ccp_act=ping_v6&ping_addr=$([injected command])
```

Old-Days

CVE-2015-1187 – Same story, different vuln

- No mention of SDK in any writeups, but…
- ncc is part of wolf cameo framework of tools
  - Used in various routers from different vendors
- ncc binary in D-Link DIR808L firmware has lots of mentions of “RT288x_SDK”
  - Indicates inclusion in SDK of at least one Realtek chipset
- So...count it?

Old-Days

Broadcom SetConnectionType Format String Vuln

- Format string vuln in NewConnectionType field of SetConnectionType command
  - Discovered by Leon Juranic and Vedran Kaji in 2013
  - Also discovered by HD Moore as part of his UPnP research?
  - No CVE in the advisory so hard to say for sure
- Vulnerable binary used in devices from Broadcom, TP-Link, Netgear, and many many more
  - 116+ different router models affected according to Ars

Old-Days

Broadcom SetConnectionType Format String Vuln

```
<SOAP-ENV:Body>
  <m:SetConnectionType
xmlns:m="urn:schemas-upnp-org:service:WANIPConnection:1" as="">
    <NewConnectionType>[format string]</NewConnectionType>
  </m:SetConnectionType>
</SOAP-ENV:Body>
```

Old-Days

Broadcom SetConnectionType Format String Vuln

- Researchers got no response from vendors
- Started popping up in botnets 5 years after disclosure
- BCMUPnP_Hunter botnet grew to over 100,000 devices using just this vuln
- Not on CISA KEV (due to lack of CVE?)

0-Days

0-Days

On Semiconductors Quantenna Chipsets

- Quantenna used to make wifi chipsets
  - Bleeding edge, first-to-market type chip maker
  - Very popular in mid-to-late 2010’s telecom equipment like AT&T and DirecTV
  - Used in Arris, Netgear, Cisco, Zyxel, etc routers and access points
- Bought by On Semiconductor in 2019
  - Became...less popular and widely used
  - Still used by a couple telecoms in Europe/Asia
  - Still making new chipsets like QCS-AX2 Wifi 6E line

0-Days

On Semiconductors Quantenna Chipsets

- Full SoC
  - ARC architecture
  - “ruby-linux” and “topaz-linux”
  - Kernel 2.6.30
- HTTP(S), UPnP, Telnet, etc
- Remote qcsapi port??

2. Open Ports

Open ports were analyzed using netstat utility.

Port	Descrption	Possible Actions
TCP 23	Telnet default port	Telnet access Services Control
TCP 80	WebUI HTTP port	WebUI access WebUI over HTTPS Services Control
TCP 111	Portmapper/sunrpc port used in remote qcsapi protocol	Open only if qcsapi rpc enabled.
TCP 443	WebUI HTTPS port	WebUI access WebUI over HTTPS Services Control
TCP 970	Remote qcsapi port	Open only if qcsapi rpc enabled.
TCP 3490	Qevent server port	Services Control
TCP 49152	WPS UPnP port	Can’t be closed.
UDP 68	dhcpclient IPv4 port	Services Control
UDP 111	Portmapper/sunrpc port used in remote qcsapi protocol	Open only if qcsapi rpc enabled.
UDP 723	Qcsapi port	Open only if qcsapi rpc enabled.
UDP 1900	WPS UPnP port	Can’t be closed.
UDP 49300	SCS BRCM IOT port used in Mediaroom environment	Can't be closed for now

Quantenna Confidential - Under NDA Only

https://onsemineworg.my.salesforce.com/sfc/p/6g000003tQ26/a/dK0000003VKD/6TQtYyJxKG4Un4f1hAmAi20.mqk4xu0EVhY6loFkSTQ

0-Days

QCSAPI

- Quantenna Configuration and Status API
- Used to get info and make config changes to wifi
  - Hundreds of commands!
    - get_firmware_version, flash_image_update
    - enable_interface
    - get_SSID, get_PSK
    - get_wps_ap_pin, enable_wps_push_button
- https://github.com/openwrt-es/qcsapi

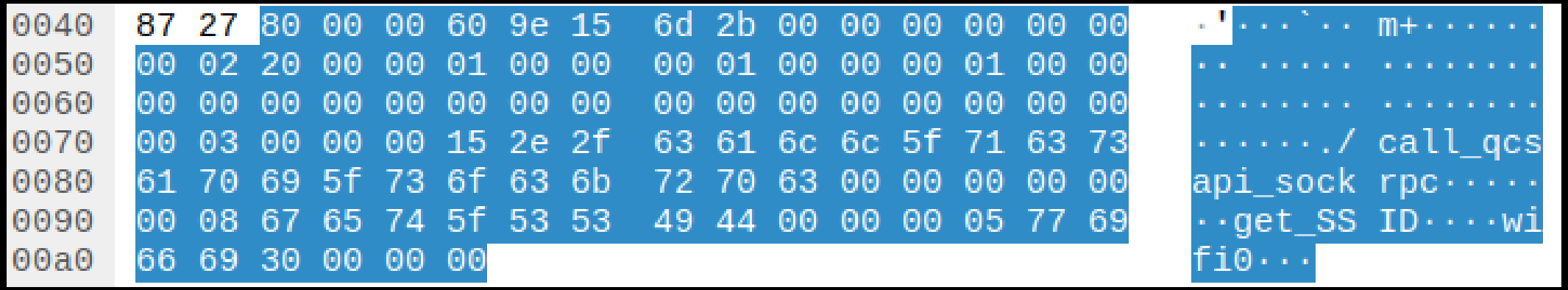
0-Days

QCSAPI – RPC Service

- RPC service for sending qcsapi commands over TCP, UDP, or PCIe
- Spawns on different ports
  - Registers with portmapper service (libtirpc)
  - Clients connect to portmapper port first to look up libtirpc port
  - Server returns port for listening rpc service
  - Makes internet-wide scanning tricky
- Some data followed by qcsapi command and argument strings
  - [4-byte len][command][4-byte len][arg1][4-byte len][arg2]...

0-Days

QCSAPI – RPC Service



```
0040   87 27 80 00 00 60 9e 15   6d 2b 00 00 00 00 00 00    .'...`.. m+......
0050   00 02 20 00 00 01 00 00   00 01 00 00 00 01 00 00    .. ..... ........
0060   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00    ........ ........
0070   00 03 00 00 00 15 2e 2f   63 61 6c 6c 5f 71 63 73    ......./ call_qcs
0080   61 70 69 5f 73 6f 63 6b   72 70 63 00 00 00 00 00    api_sock rpc.....
0090   00 08 67 65 74 5f 53 53   49 44 00 00 00 05 77 69    ..get_SS ID....wi
00a0   66 69 30 00 00 00                                    fi0...
```

0-Days

QCSAPI – run_script

- run_script command seems like a good place to look for vulns!
- Looking at call_qcsapi.c shows whitelist of scripts you can run

```
781
782     char *support_script_table[] = {
783                     "stop_test_packet",
784                     "send_test_packet",
785                     "set_test_mode",
786                     "set_tx_pow",
787                     "send_cw_signal",
788                     "stop_cw_signal",
789                     "send_cw_signal_4chain",
790                     "show_test_packet",
791                     "transmit_file",
792                     "remote_command",
793                     "router_command.sh"
794     };
795
```

0-Days

QCSAPI – run_script

- Scripts can be found in firmware
- 7 distinct cmd inj vulns across the scripts, and this was the first place I looked
- transmit_file as an example:

```
5cd34967dr:~/research/routers/quantenna$ cat transmit_file
#! /bin/sh

if [ $# -ne 2 ]
then
    echo "Usage: transmit_file <image name> <host IP address>"
    exit 1
fi

tftp -g $2 -r $1 -l /tmp/$1
```

0-Days

QCSAPI – telnet

- Telnet service is passwordless by default
  - Vendors must manually add passwd or chpasswd pkgs
  - “By default, the security options are disabled to allow for ease of our customer product development.” ~ onsemi
- Telnet is enabled by default
  - Vendors must disable telnet via build config flags
  - To re-enable, you just run the S42inetd script
  - ...Don’t we have a vuln that let’s us run any command we want?

Demo

0-Days

DEMO - Target

- Target is ASUS RT-AC87U
  - Old, but Quantenna chips hard to come by these days
    - Vulns still exist in newer chipsets, I promise
  - Device is interesting
    - Two distinct CPU’s networked together
      - Broadcom BCM4709A0 (main router OS, ARM, IP is configured by user)
      - Quantenna QT3840BC (wifi OS, ARC, IP is 169.254.39.129)
    - Router OS communicates with wifi OS over TCP via qcsapi_rpc calls
- Manually setting your IP to 169.254.39.x allows you to route to Quantenna chip
- Telnet was disabled in build config, as per best practice instructions...

0-Days

DEMO - Target

- [play demo video now]

0-Days

Response from onsemi

- Vulns disclosed to onsemi and were acknowledged on 2025/04/11
- Onsemi published a best practices guide for their customers as a response
  - Pushes security ownership/responsiblity onto their customers
  - Advises telnet be disabled (it was, I re-enabled it using the vuln)
  - Assures that "set_tx_pow" will not exceed FCC limits...what?
- Onsemi requested to include a link to that guide in lieu of actual fixes, so...here it is...
  - https://community.onsemi.com/s/article/QCS-Quantenna-Wi-Fi-product-support-and-security-best-practices
- All 8 CVE advisories can be found here: https://takeonme.org/cve/#2025-disclosures

Conclusions

Conclusions

Secure by Default

- Don’t make your customers secure your stuff for you
  - Makes things too complicated? Fix it with more documentation, not less security
- Secure it upfront or end up chasing botnets for the rest of time
  - Once a vuln is out, it’s much harder to put it back in
- Maintain exhaustive SBOMs for your products
  - Include standard and non-standard things
  - Have a plan for keeping up with updates
  - Regular audits for versions, running services, etc
  - STOP SILENT PATCHING! Only makes tracking updates harder

Conclusions

Helpful Links

- https://wikidevi.wi-cat.ru/
  - Great resource for info on chipsets and the devices that use them
- https://www.cisa.gov/known-exploited-vulnerabilities-catalog
  - Use the Vendor/Project filters to keep an eye on your stuff
- https://github.com/HeadlessZeke
- @HeadlessZeke@infosec.exchange

KEYSIGHT

Thank you


KEYSIGHT