

# **A SMART LIGHT HACKING JOURNEY**

# ABOUT US

## KHALED NASSAR

- Security Engineer
- Avid CTF player

SPoTLess

 notkmhn

 @notkmhn

## TOM CLEMENT

- Organiser
- Sr. embedded software
- Owner



Ultimaker

 tjclement

 @Tom\_Clement

CURIOUS  
SUPPLIES

# A QUICK LOOK AHEAD

INTRODUCTION AND PRIOR WORK

ESP8266 - INITIAL INVESTIGATION

BK7231 - NEW CHIP, NEW VULNS

VULNERABILITY DETAILS AND EXPLOITATION



# INSPIRATION



## HACK42 FLASHING PARTY

OVER HACK42 NIEUWS AGENDA MEEDOEN TOUR STATS **WIKI** CONTACT SHOP

### Home Automation, Domotica en Internet of Things/LSC Smart Connect Slimme Stekker

ACCOUNT AANVRAGEN AANMELDEN

Zoeken  pagina [overleg](#) [brontekst bekijken](#) [geschiedenis](#)

OK Zoeken

#### Actualiteiten

- Recente wijzigingen

#### Verhuizingdingen

- Maken Pand4.1

#### Activiteiten & projecten

- Activiteiten
- Projecten
- Kalender
- Activiteit Toevoegen
- Project Toevoegen

#### Faciliteiten

- Howto Hack42
- Het gebouw
- Gereedschap
- Museum

#### Diversiteiten

- Huisregels
- Deelnemers
- Wiki-nieuwkomers
- Wiki powerusers
- Pers & media
- Sponsors
- Handige links

## Flashen

De action slimme stekker is veel slimmer te maken en minder afhankelijk van tuya connect

De makkelijkste manier is [github:tuya-convert](#) op je laptop/rasberry pi/device met wifi te installeren. Volg de handleiding netjes en hark de software bij elkaar als je iets anders dan debian draait om het goed te laten werken.

Na het inpluggen van de stekker een paar seconde de push button ingedrukt houden en je hoort een klikje en de stekker begint te knippen. Dan is die in flash modus, tik 'y' en/of ENTER om het flashen te starten.

Aan het einde van de flash het curl commando naar flash3 draaien en er staat tasmota op. (duurt ongeveer 17 seconden, heb wat geduld).

Nadat er [tasmota](#) op staan kun je met je telefoon / ... inloggen (telefoon is handig!) op tasmota-<nummer> en een wifi netwerk kiezen.

Als je op je DHCP server, via DNS (of met nmap) het device kunt vinden als tasmota-<nummer>.

Hint: meestal werkt [http://tasmota-<nummer>](#) wel in je browser.

Kies dan voor "console" en paste de volgende regel:

```
reset 5
```

Wacht even tot de module geherstart is en paste:

# INSPIRATION



## HACK42 FLASHING PARTY

OVER HACK42 NIEUWS AGENDA MEEDOEN TOUR STATS **WIKI** CONTACT SHOP

### Home Automation, Domotica en Internet of Things/LSC Smart Connect Slimme Stekker

[github.tuya-convert](#) PUNT AANVRAGEN AANMELDEN

Doorzoek Hack42  pagina [overleg](#) [brontekst bekijken](#) [geschiedenis](#)

#### Actualiteiten

- Recente wijzigingen

#### Verhuizingdingen

- Maken Pand4.1

#### Activiteiten & projecten

- Activiteiten
- Projecten
- Kalender
- Activiteit Toevoegen
- Project Toevoegen

#### Faciliteiten

- Howto Hack42
- Het gebouw
- Gereedschap
- Museum

#### Diversiteiten

- Huisregels
- Deelnemers
- Wiki-nieuwkomers
- Wiki powerusers
- Pers & media
- Sponsors
- Handige links

## Flashen

De [action slimme stekker](#) is veel slimmer te maken en minder afhankelijk van [tuya connect](#).

De makkelijkste manier is [github.tuya-convert](#) op je laptop/raspberry pi/device met wifi te installeren. Volg de handleiding netjes en hark de software bij elkaar als je iets anders dan debian draait om het goed te laten werken.

Na het inpluggen van de stekker een paar seconde de push button ingedrukt houden en je hoort een klikje en de stekker begint te knippen. Dan is die in flash modus, tik 'y' en/of ENTER om het flashen te starten.

Aan het einde van de flash het curl commando naar flash3 draaien en er staat tasmota op. (duurt ongeveer 17 seconden, heb wat geduld).

Nadat er [tasmota](#) op staan kun je met je telefoon / ... inloggen (telefoon is handig!) op [tasmota-<nummer>](#) en een wifi netwerk kiezen.

Als je op je DHCP server, via DNS (of met nmap) het device kunt vinden als [tasmota-<nummer>](#).

Hint: meestal werkt [http://tasmota-<nummer>](#) wel in je browser.

Kies dan voor "console" en paste de volgende regel:

```
reset 5
```

Wacht even tot de module geherstart is en paste:

# PREVIOUS WORK - TUYA-CONVERT

ct-Open-Source / [tuya-convert](#) Public

A collection of scripts to flash Tuya IoT devices to alternative firmwares


MIT license

3.6k stars 425 forks

Star Watch

Code Issues 149 Pull requests 15 Discussions

master

 **kueblc** Release v2.4.5 ... on 26 Jan 2021 400

[View code](#)

README.md

## TUYA-CONVERT

## Smart Home - Smart Hack

### Wie der Weg ins digitale Zuhause zum Spaziergang wird

Michael Steigerwald



### BIG DATA (personenbezogene Daten)

Device activating information	
Activation status	Yes
Time of activation	2018-11-22 07:38:12
Last device activity	2018-11-27 09:16:04
Last update	2018-11-27 09:16:25
Online now	No
Binding user	35c3@vtrust.de
Binding APP	涂鸦智能
Latitude and longitude	51.397840, 12.405506
Geographic position	Leipzig
Channel	
Time zone	Europe/Berlin GMT+01:00

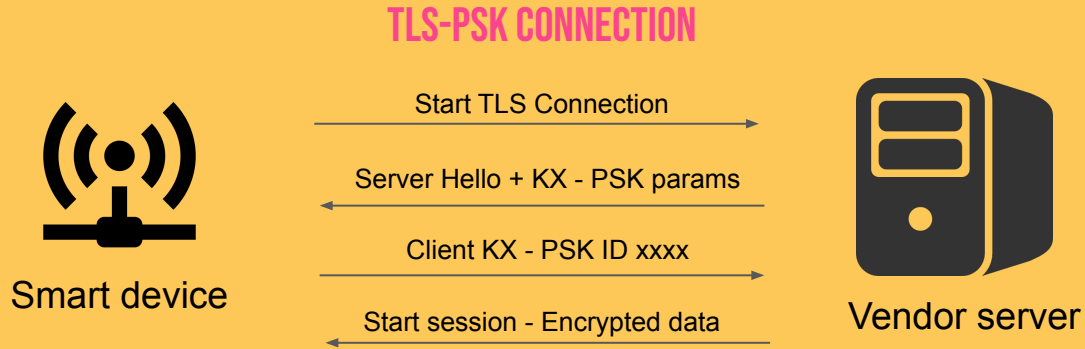


Playlists: '35c3' videos starting here / audio / related events

51 min 2018-12-28 2018-12-29 40610 [Fahrplan](#)

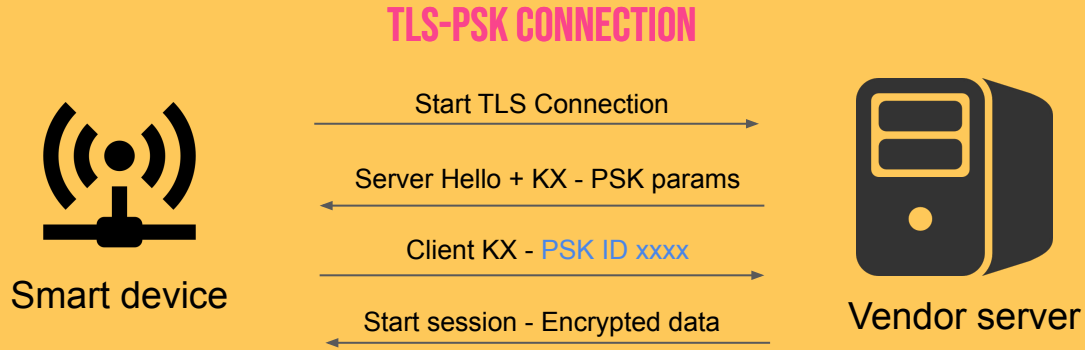
# PREVIOUS WORK - TUYA-CONVERT

## WHAT WAS THE ISSUE?



# PREVIOUS WORK - TUYA-CONVERT

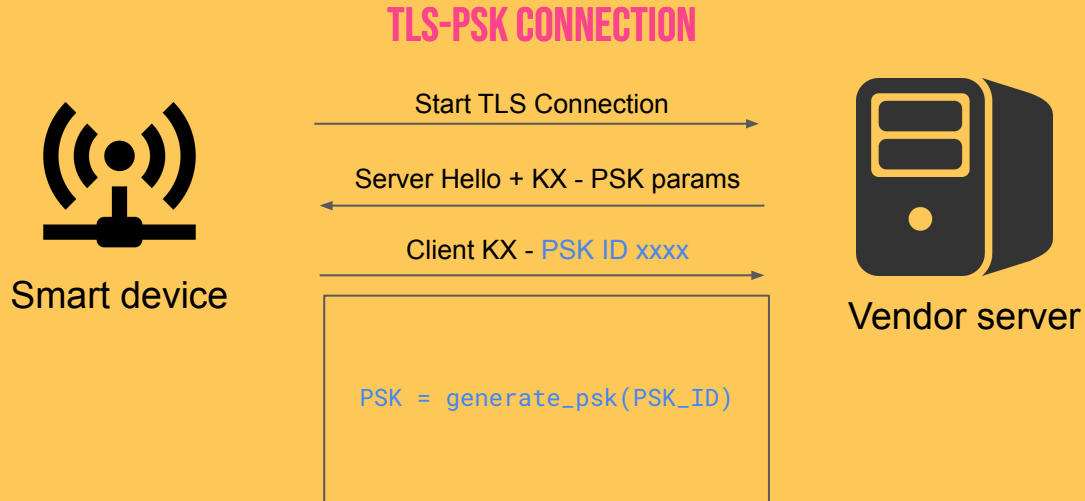
## WHAT WAS THE ISSUE?





# PREVIOUS WORK - TUYA-CONVERT

## WHAT WAS THE ISSUE?

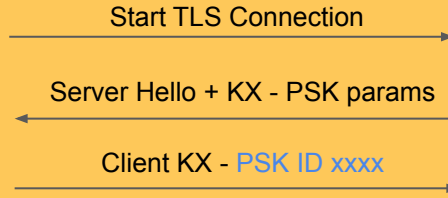


# PREVIOUS WORK - TUYA-CONVERT

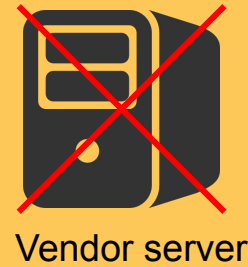
## WHAT WAS THE ISSUE?



### TLS-PSK CONNECTION

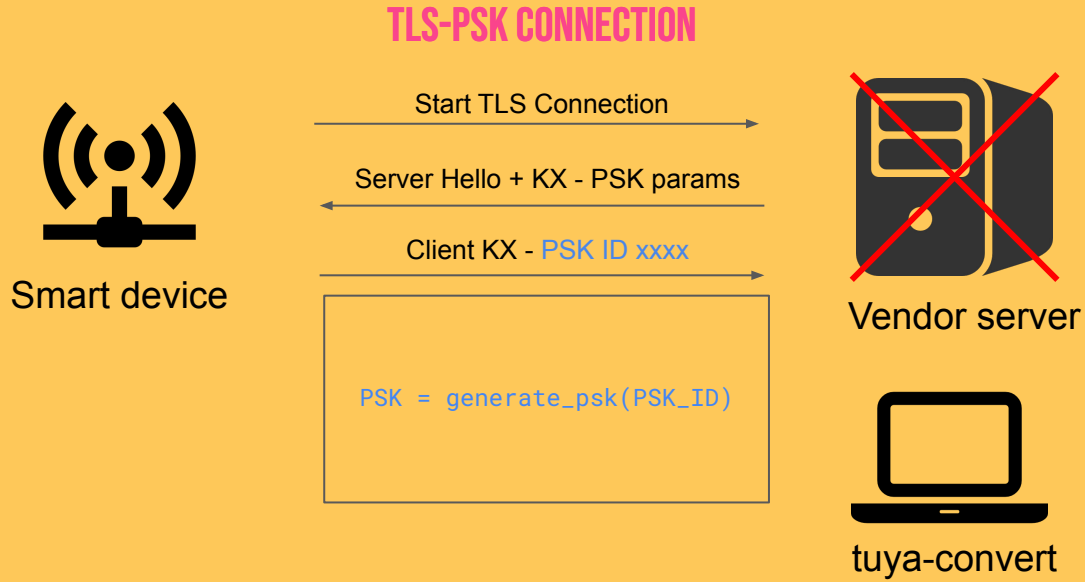


```
PSK = generate_psk(PSK_ID)
```



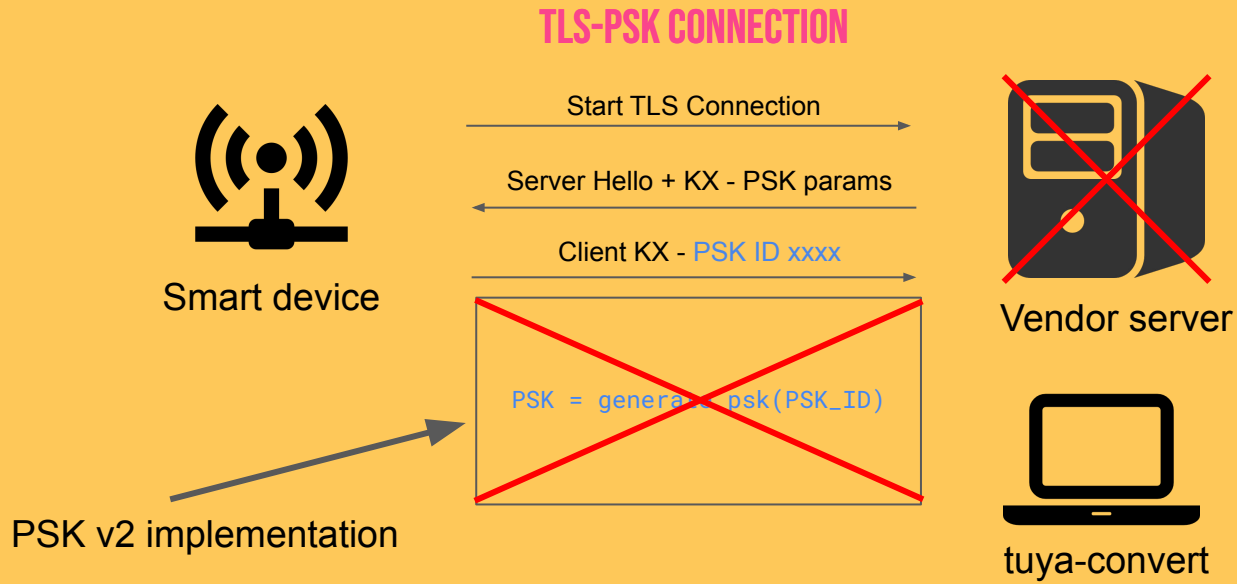
# PREVIOUS WORK - TUYA-CONVERT

## WHAT WAS THE ISSUE?



# PREVIOUS WORK - TUYA-CONVERT

## WHAT WAS THE ISSUE?



# GAMEPLAN



# GAMEPLAN

- Get the PSK
  - Overwrite
  - Leak



# GAMEPLAN

- Get the PSK
  - Overwrite
  - Leak
- Downgrade to vulnerable protocol version



# GAMEPLAN

- Get the PSK
  - Overwrite
  - Leak
- Downgrade to vulnerable protocol version
- Get code execution on the light bulb





# ESP8266 - INITIAL INVESTIGATION

# DUMPING FIRMWARE

ESPTOOL



`esptool.py [.] read_flash`



Flash image

# LOADING BINARY INTO GHIDRA



Raw binary



esp-bin2elf



Ghidra + Xtensa processor module

# IDENTIFYING KNOWN FUNCTIONS

esp8266\_2.1.1\_smartconfig.effFunctions.csv - LibreOffice Calc

A	B	C	D
1	call_user_start	0x040100004	a71284cd47b05da72857e16c49a2b03a0c30465
2	DebugExceptionVector	0x040100010	70ae387eae7e4d539276a722415b0ca0af6895e
3	NMIEExceptionVector	0x040100020	da39a3ee6eb4b0cd325b0ef95601890ad80079
4	UserExceptionVector	0x040100050	3685e3ad60214544ea60419202351915233006
5	FUN_40100098	0x040100098	6693bbe9c245148271b6794ac754d266d7c8ab
6	pinHeaderBlockIntoUsedList	0x040100040	ed29002033915122838954ab0ca29ea9997724d2
7	pinHeaderBlockFromUsedList	0x040100104	b02b02c3a7537510260a75cc7386ca65890b9de9
8	FreeFree	0x04010014c	aa6c9246e90ed0b47b531728253eb48494d4c
9	free	0x04010010e	2090d1774774487b8815dea7c26aa95cd99
10	xPortGetFreeHeapSize	0x04010018f	fb20ata87e11ae285ac4898e40534c22ab4a3
11	xPortDefineHeapRegions	0x040100208	68d00e938e5281d7b1b14371f94a270c089b
12	xPortMalloc	0x0401002a8	cfc7159002833090a2c57a2ab45cd7217bc64
13	xPortCalloc	0x040100410	64f58f696a3a35784ac414772cae695953612
14	xPortZalloc	0x040100448	ca45343a75bc9d27e1bed189b2e29200563bd06
15	zalloc	0x040100464	add9e3de4bf243e12c12476c635ab05d90078
16	calloc	0x04010047c	cd00c08c1b7b143818136e12c7972462226e9
17	xPortRealloc	0x040100498	95330d175bf1b6804255be35231e407e1ca64da
18	realloc	0x0401004fc	3069a663717ebdf5afe62a1d7a3efc11599018e7
19	malloc	0x040100514	78bc4e4643cd03c8db847b448467e3b7807
20	Socketsrv	0x040100538	6d76995c52ea8986e513514516c5e9442c30ab5
21	xPortSysTickHandle	0x040100568	1d98e4dec445328c378e09534eac33789b4b09
22	xPortEnterCritical	0x040100584	523-c2a8113543aa15b110c58af6c4ba9a3ca541
23	xPortExitCritical	0x0401005ec	ab78495cbec4e9ff8db6ac58d3c458e868695d4
24	PendSV	0x040100648	3d6f635b490a0992d4064b5acd9d7116a2e28
25	xPortTSInitLock	0x040100688	99e47423c5435950ad650e05cd051743c9d1e
26	xPortTSInitUnlock	0x0401006b8	d0071371da9578922adb07b0a6d7348e55efc
27	PortDisableInt_NoHost	0x0401006e0	aa8b9952bc2a7407e1e462207f8d8c730e00c
28	xPortHandler	0x040100708	e286411131737810c1c69e117a130114
29	FUN_401000794	0x040100794	491820992a520e3006434934ac703b95989d4
30	FUN_401000824	0x040100824	d08a4495089540779c6711e59531112327a
31	xTaskIncrementTick	0x040100878	e3e2c384798a0ec86dd45837b5441b0c2
32	xTaskSwitchContext	0x04010090e	42bc90cddcd39dabb6e18b2ab0d204e824b7f
33	xTaskPriorityDisinherit	0x040100a4a	2243466a3d4327143b35f1a35acc39ebdb6f
34	xTaskStackSizeAlign	0x040100a8a	2137be04429eb303aleb17249951c50a29d
35	uxQueueMessagesWaitingFromSR	0x040100a0c	5b39631db4d29a7e8997aac049853cd729e64c
36	uxListInsertEnd	0x040100b04	485b7ae7ccc29510ca47e859a0b08d6904d1d
37	uxListRemove	0x040100b1c	592c3a4b763792a455882b0d404db1b74ad5b
38	SPIWrite	0x040100b4c	722c1807b43573434da618ba9d407064993
39	SPIRead	0x040100c08	665bcc23d5409fa5723186ceebb157c5a2d3603
40	SPIEraseSector	0x040100c38	04da7149506b2c2db25ada543781e0c8464d95
41	Cache_Read_Disable_2	0x040100c9c	75aea3004c3bd18db8b406731bc8156154acd5
42	Cache_Read_Enable_2	0x040100cd8	laba2a74e4542474037138999ab14845a5a1
43	spi_flash_get_id	0x040100d1c	ec699511ca7a76ae056d1b0515195267961752

userfirm\_new.allFunctions.csv - LibreOffice Calc

A	B	C	D
127	fixsumdsfi	0x04000cd00	6b2542754c49750675107ba79907b6b9b620907
128	truncdfs2	0x04000cd5c	298a67393b14d1551a04b531b955ed6b579d3a
129	extenddsf2	0x04000cd6c	14a82ce9e62b25e20ab945100b38e81cd9609
130	divd3	0x04000ce60	2983a9e524b442362e6f52206e7eadb9f78c88
131	udivd3	0x04000d310	77b739020a101b175324fb20603b7d4d05093f
132	umodsd3	0x04000f770	00089a5e4d426572011322ba20786904d4cc908
133	dvsd3	0x04000a88	628b29568b34857c2d7c52b7178e68b6bae6
134	umulsd3	0x04000d4f0	45acc023ace182a6625982ab4542541d5638a
135	FUN_4000d38	0x04000d38	5dfec73d6e465a368a9aca5328e5a4e466a07
136	FUN_4000de4	0x04000de4	e67204c17d0c008be61b929ca21820999d98f
137	memcmp_actual	0x04000de8a	c683d418952837e4187680b8da8a56led02ea
138	memcmpv_actual	0x04000f48	8197e51c8d9c9e987dfcd3f53a9e0a026706c8a
139	memcpyv_actual	0x04000e04c	4c1c5d52f70e388e8a01d72987741407956
140	memsetv_actual	0x04000e190	002e8918d36490c71d9c48e6c2b94835cb3d875
141	strcatv_actual	0x04000e1e0	d8ed0bc4e45a322ba65908851b0e93c75c02
142	udivs3	0x04000e21c	af109df9e8b6e8ab50dcd771ed8187191add6
143	umods3	0x04000e268	265cc609226204995c03da0a30d08aa1e753
144	floatsysf!	0x04000e2a4	51e86acc0c75851b698cac744584cd37b19e
145	floatsysf!	0x04000e2ac	d72e1b13559109054868390d9607316912897
146	floatsysf!	0x04000e2e8	8789e76508908376789480867782076499a3c
147	floatsysf!	0x04000e2b0	bb74ad669cd112988e1c7e9d989181708489
148	FUN_4000e324	0x04000e324	2137b0444e29b3d3aleb172b4951c50a29d
149	entry	0x040100004	30cc4abb36bc76481156a9d3407c6a32117
150	FUN_40100098	0x040100098	f693bbe9c245148271b6794ac754d266d7c8ab
151	FUN_401000e0	0x0401000e0	ed2908203915102b3885ab2c2dea999e724a
152	FUN_40100104	0x040100104	bd9e2eed3d7576500a75cc736e9a5989b9b9
153	free	0x04010014c	920631990c250a7b753d8a365a450c0cd68
154	free	0x040100170	8197e51c8d9c9e987dfcd3f53a9e0a026706c8a
155	FUN_40100118	0x040100118	8197e51c8d9c9e987dfcd3f53a9e0a026706c8a
156	FUN_40100128	0x040100128	8197e51c8d9c9e987dfcd3f53a9e0a026706c8a
157	malloc	0x0401002a8	50082b68e98d1c9444321d8ebc4a02b66c72b
158	zalloc	0x040100410	64f58f696a3a35784ac414772cae695953612
159	zalloc	0x040100448	ca45343a75bc9d27e1bed189b2e29200563bd06
160	calloc	0x040100464	add9e3de4bf243e12c12476c635ab05d90078
161	calloc	0x04010047c	cd00c08c1b7b143818136e12c7972462226e9
162	realloc	0x040100498	95330d175bf1b6804255be35231e407e1ca64da
163	FUN_40100534	0x040100534	11772af5c9a6b0e5f985c9e02e087f66e491

Confirm import

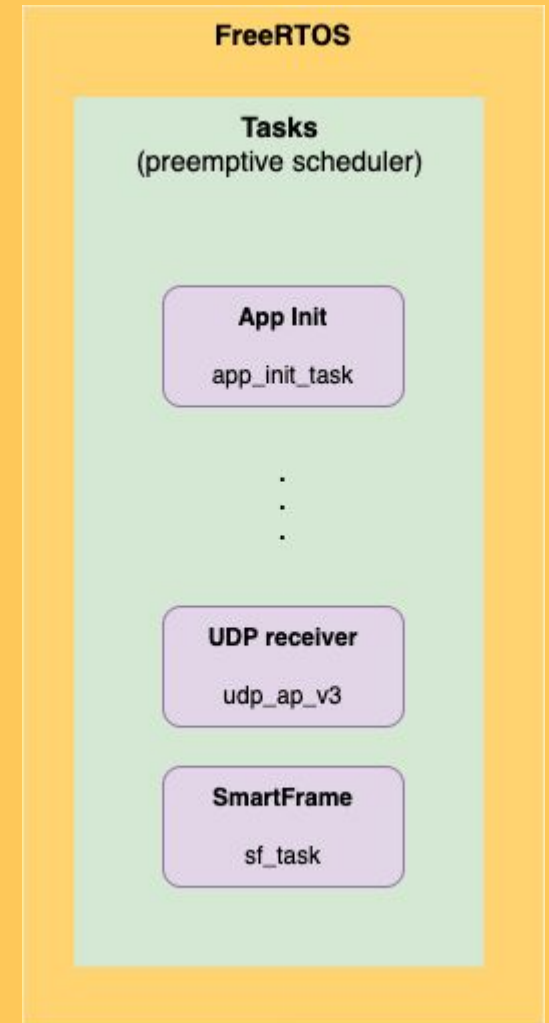


Match FUN\_40107144 to \_\_mulsd3 with 1% difference? This function was already imported to an equal or better fitting function.

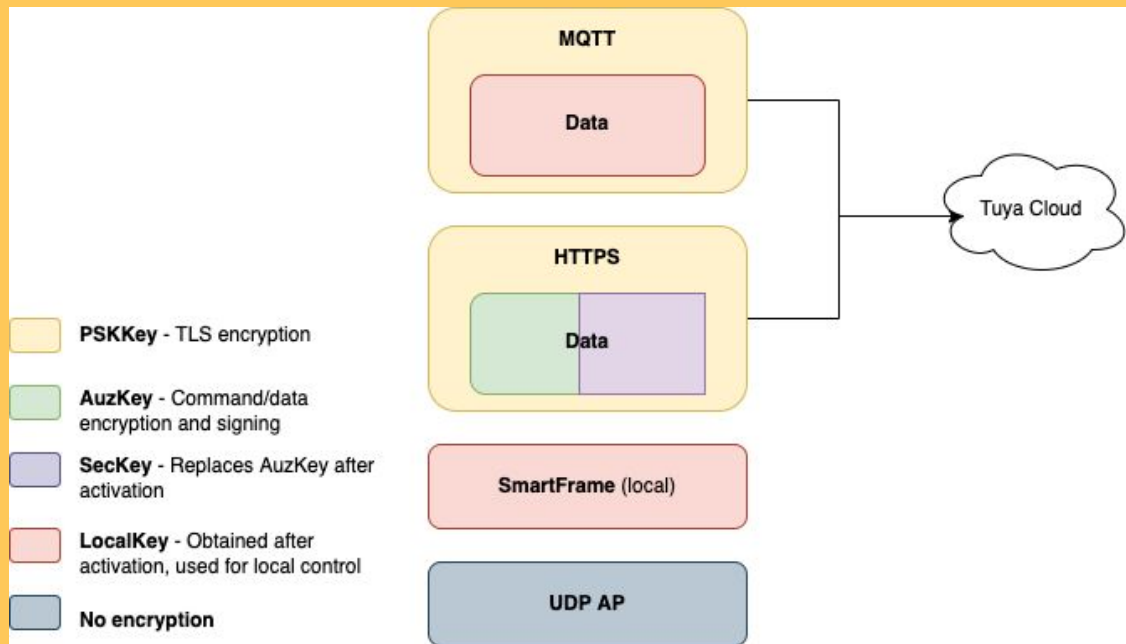
Yes No

# TUYA STACK

- FreeRTOS tasks
  - Initialization
  - "Smart config"
  - MQTT
  - Peripheral control
- HAL
  - Device configuration
  - WiFi
  - Bluetooth
  - Other utilities



# GETTING ROOT



# ESP8266 - VULNERABILITIES

# BAD RANDOM GENERATION™

```
undefined4 get_random_string(uint unused, char *dest, uint max_length)
{
    int clock_rand;
    int adc_rand;
    uint length;
    clock_rand = system_rtc_clock_cali_proc_inner();
    adc_rand = wrapper_r_rand_from_adc();
    clock_rand = __umodsi3(clock_rand + adc_rand, 0x15f);
    length = 0x15eU - clock_rand;
    if (max_length < 0x15eU - clock_rand) {
        length = max_length;
    }
    memcpy_actual(dest, s_BAohbmd6aG91IFR1eVjaG5vbG9SBUwEw_3ffe9310, length);
    return 0;
}
```



# BAD RANDOM GENERATION™

```
undefined4 get_random_string(uint unused, char *dest, uint max_length)
{
    int clock_rand;
    int adc_rand;
    uint length;
    clock_rand = system_rtc_clock_cali_proc_inner();
    adc_rand = wrapper_r_rand_from_adc();
    clock_rand = __umodsi3(clock_rand + adc_rand, 0x15f);
    length = 0x15eU - clock_rand;
    if (max_length < 0x15eU - clock_rand) {
        length = max_length;
    }
    memcpy_actual(dest, s_BAohbmd6aG91IFR1eVjaG5vbG9SBUwEw_3ffe9310, length);
    return 0;
}
```

# BAD RANDOM GENERATION™

```
undefined4 get_random_string(uint unuse
{
    int clock_rand;
    int adc_rand;
    uint length;
    clock_rand = system_rtc_clock_calibrat
    adc_rand = wrapper_r_rand_from_adc();
    clock_rand = __umodsi3(clock_rand + a
    length = 0x15eU - clock_rand;
    if (max_length < 0x15eU - clock_rand)
        length = max_length;
}
memcpy_actual(dest, s_BAohbmd6aG91IFR1
return 0;
}
```

```
▼ Transport Layer Security
  ▼ TLSv1.2 Record Layer: Handshake Protocol: Client Hello
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 62
  ▼ Handshake Protocol: Client Hello
    Handshake Type: Client Hello (1)
    Length: 58
    Version: TLS 1.2 (0x0303)
  ▶ Random: 42416f6842416f68626d6436614739314946523165566a61...
    Session ID Length: 0
    Cipher Suites Length: 4
  ▶ Cipher Suites (2 suites)
    Compression Methods Length: 1
  ▶ Compression Methods (1 method)
    Extensions Length: 13
  ▶ Extension: max_fragment_length (len=1)
  ▶ Extension: encrypt_then_mac (len=0)
  ▶ Extension: extended_master_secret (len=0)
```

0000	6e 13 d1 9c 71 dd c4 4f 33 b8 88 8b 08 00 45 00	n...q..0 3.....E.
0010	00 6b 00 10 00 00 ff 06 53 12 0a 2a 2a 16 0a 2a	.k.....S..***..*
0020	2a 01 bf 6b 01 bb 00 00 19 70 20 48 7a fb 50 18	*..k.....p Hz.P.
0030	11 1c 47 d9 00 00 16 03 03 00 3e 01 00 00 3a 03	.G.....>.....
0040	03 42 41 6f 68 42 41 6f 68 62 6d 64 36 61 47 39	·BAohBAo hbmd6aG9
0050	31 49 46 52 31 65 56 6a 61 47 35 76 62 47 39 53	1IFR1eVj aG5vbG9S
0060	42 00 00 04 00 ae 00 ff 01 00 00 0d 00 01 00 01	B.....
0070	02 00 16 00 00 00 17 00 00	.....

# BAD RANDOM GENERATION™

```
undefined4 get_random_string(uint unuse
{
    int clock_rand;
    int adc_rand;
    uint length;
    clock_rand = system_rtc_clock_calibrat
    adc_rand = wrapper_r_rand_from_adc();
    clock_rand = wrapper_r_rand_from_adc();
```

```
Transport Layer Security
  TLSv1.2 Record Layer: Handshake Protocol: Client Hello
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 62
  Handshake Protocol: Client Hello
    Handshake Type: Client Hello (1)
    Length: 58
    Version: TLS 1.2 (0x0303)
  Random: 42416f6842416f68626d6436614739314946523165566a61...
    Session ID Length: 0
    Cipher Suites Length: 4
  Cipher Suites (2 suites)
    Compression Methods Length: 1
  Compression Methods (1 method)
    Extensions Length: 13
  Extension: max_fragment_length (len=1)
  Extension: encrypt_then_mac (len=0)
  Extension: extended_master_secret (len=0)
```

This function which returns a prefix of a long hardcoded string is provided as the RNG function to be used by mbedTLS through an invocation of `mbedtls_ssl_conf_rng`. It is then used for all operations which require randomness for TLS purposes. Said operations include parameter generation for key exchange algorithms. For example, the ClientRandom in the TLS Client Hello message as shown in traffic, as well as ClientSecret for DH key exchange. Meaning that TLS MITM is possible by leveraging the predicted client-side secrets (with high frequency of success) to derive the TLS Pre-master Secret.

It is not exploitable since KX-based suites are not used at all, only TLS-PSK is.

```
88 8b 08 00 45 00  n...q..0 3...E.
0a 2a 2a 16 0a 2a  .k.....S...**..*
20 48 7a fb 50 18  *..k.... .p Hz.P.
3e 01 00 00 3a 03  .G.....>...:
6d 64 36 61 47 39  BAohBAo hbmd6aG9
35 76 62 47 39 53  1IFR1eVj aG5vbG9S
00 0d 00 01 00 01  B.....
.....
```

# STACK + HEAP BUFFER OVERFLOW

```
[N]mqtt_client.c:603 gw wifi stat is:3  
Fatal exception (9):  
epc1=0x401000e3  
epc2=0x00000000  
epc3=0x4025eeb0  
epcvaddr=0x40006c6e  
depc=0x00000000  
rtn_add=0x401003d4
```

```
Free Heap Size: 27112
```

```
Stack Point: 3fff5900
```

```
3fff5900: 40226890 80000000 00006a20 3ffefe44  
3fff5910: 00000023 3fff5940 3fff1b2c 00000000  
3fff5920: 00000000 40108800 00000023 401004a6  
3fff5930: 00000000 40108800 00000023 4025eb65  
3fff5940: 00000045 40107ec8 00000001 00000000  
3fff5950: 40108800 00000000 00000020 4025f260  
3fff5960: 40107f80 40107ec8 00000022 00000008  
3fff5970: 00000000 00000024 00000024 00000000
```

# STACK + HEAP BUFFER OVERFLOW

```
[N]mqtt_client.c:603 gw wifi stat is:3
Fatal exception (9):
epc1=0x401000e3
epc2=0x00000000
epc3=0x4025eeb0
epcvaddr=0x40006c6e
depc=0x00000000
rtn_add=0x401003d4

Free Heap Size: 27112
```

Stack buffer overflow in the smart\_config task, which is implemented as an infinite loop and never returns, hence no IP control. No interesting and usable data on the stack either.

Heap buffer overflow vulnerability in the same smart\_config task. Manipulating heap blocks is possible, but could not exploit it further due to input constraints (no null bytes allowed in trigger payload).

## "DYNAMIC ANALYSIS" - RAM DUMP

- Reset to bootrom
- esptool to dump RAM

# "DYNAMIC ANALYSIS" - RAM DUMP

- Reset to bootrom
- esptool to dump RAM

```
BlockLink_40107720                                XREF[2]:  3ffefe2c(*), 3fffaab0(*)
40107720 50 77 10      BlockLink
      40 30 00
      00 80 90 ...
┌── 40107720 50 77 10 40      BlockLin...BlockLink_40107750      pxNextFreeBl... =      XREF[2]:  3ffefe2c(*), 3fffaab0(*)
├── 40107724 30 00 00 80      size_t      80000030h      xBlockSize
├── 40107728 90 68 22 40      char *      s_user_app_40226890      file      = "user_app"
└── 4010772c 00 00 00 00      uint32_t    0h      line
40107730 30      ??      30h  0
40107731 9d      ??      9Dh
40107732 10      ??      10h
40107733 40      ??      40h  @
```

# "DYNAMIC ANALYSIS" - RAM DUMP

- Reset to bootrom
- esptool to dump RAM
- Debugging over JTAG with gdb

```
BlockLink_40107720                                XREF[2]: 3ffefe2c(*), 3fffaab0(*)
40107720 50 77 10      BlockLink
      40 30 00
      00 80 90 ...
├── 40107720 50 77 10 40      BlockLin...BlockLink_40107750      pxNextFreeBl... = XREF[2]: 3ffefe2c(*), 3fffaab0(*)
├── 40107724 30 00 00 80      size_t          80000030h      xBlockSize
├── 40107728 90 68 22 40      char *          s_user_app_40226890      file = "user_app"
├── 4010772c 00 00 00 00      uint32_t        0h          line
40107730 30          ??          30h 0          ? -> 40109d30
40107731 9d          ??          9Dh
40107732 10          ??          10h
40107733 40          ??          40h @
```

notkmhn / binutils-gdb-xtensa Public  
forked from jcmvbkbc/binutils-gdb-xtensa

binutils and GDB for xtensa

GPL-2.0 and 3 other licenses found

0 stars 7 forks

Star Notifications

<> Code Pull requests Actions Projects Wiki Security ...

master Go to file

This branch is 1 commit ahead, 4653 commits behind jcmvbkbc:master. Contribute

notkmhn Changes to use gdb with esp8266-openocd on Apr 8, 2021 101,385



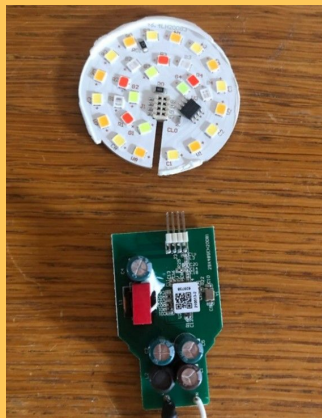
## SUMMARY

- Found a few bugs that weren't easily exploitable
- RAM dumping using esptool for static analysis with Ghidra
- Debugging worked somewhat
- Learned a lot about Tuya's stack

A FEW  
MOMENTS LATER


A red title card with white text. The text reads "A FEW MOMENTS LATER" in a bold, sans-serif font. The background is a solid red color with faint, repeating patterns of a globe and a spiral.

# NEW CHIP, WHO DIS?



# BK7231 ARCHITECTURE

## FINDING DATASHEETS



**LCSC ELECTRONICS**  
More Asian Brands, Lower Prices

[TLV61220DBVR](#) [CL10A226MP8NUNE](#) [CL05A105KA5NQNC](#)

[All Products](#) [Manufacturers](#) [BOM Tool](#) [Popular Products](#) [Deals](#) [About LCSC](#)

[◀ Back](#) [Home](#) / Search by "bk7231"

"bk7231" did not return any results

**Tips on Searching for Parts :**

- Check spelling of part number or keywords
- Use fewer or different keywords
- Search on 1 part number at a time
- Search by [Product Category](#)

# BK7231 ARCHITECTURE

## BAIDU TO THE RESCUE



The image shows a screenshot of a Baidu search engine result page. At the top left is the Baidu logo. The search bar contains the text 'bk7231t'. Below the search bar are navigation tabs for '网页' (Web), '资讯' (News), '贴吧' (Forum), '图片' (Image), '视频' (Video), '知道' (Know), '文库' (Library), '地图' (Map), '采购' (Procurement), and '更多' (More). The search results show approximately 785,000 related results. The top result is a link to 'BK7231T处理器规格-上海博通BK7231T芯片性能 报价 规格书 ...' with a snippet of text: '2020年12月9日 BK7231T BK7231T主要技术参数如下: Wi-Fi SOC芯片,内嵌arm9E处理器。1. 符合802.11b/g/n 1x1协议&nBSP; 2. 17dBm 输出功率 3. 支持20/40 MHz带宽和STBC 4...'. Below this is a link to 'BK7231\_深圳博芯科技股份有限公司' with a snippet: '描述 特性 应用方案 BK7231是一颗2.4GHz 802.11b/g/n数据传输SoC,芯片集成了802.11b/g/n从射频到MAC层所有的软硬件功能,通用的ARM9 M CU和丰富的存储资源使得芯片可以支持各种网络协议。'. To the left of this snippet is a small image of the BK7231 chip. At the bottom right of the snippet is a link to '深圳博芯科技股份有限公司'.

Baidu 百度

bk7231t

百度一下

Q 网页 资讯 贴吧 图片 视频 知道 文库 地图 采购 更多

百度为您找到相关结果约785,000个

搜索工具

[BK7231T处理器规格-上海博通BK7231T芯片性能 报价 规格书 ...](#)

2020年12月9日 BK7231T BK7231T主要技术参数如下: Wi-Fi SOC芯片,内嵌arm9E处理器。1. 符合802.11b/g/n 1x1协议&nBSP; 2. 17dBm 输出功率 3. 支持20/40 MHz带宽和STBC 4...

一牛网论坛 百度快照

[BK7231\\_深圳博芯科技股份有限公司](#)

 描述 特性 应用方案 BK7231是一颗2.4GHz 802.11b/g/n数据传输SoC,芯片集成了802.11b/g/n从射频到MAC层所有的软硬件功能,通用的ARM9 M CU和丰富的存储资源使得芯片可以支持各种网络协议。

深圳博芯科技股份有限公司 百度快照

# BK7231 ARCHITECTURE

DATASHEET



BK7231 Data Sheet

QN40V0.6

---

---

**BK7231 Data Sheet**

---

---

# BK7231 ARCHITECTURE

## 968E-S (ARM9 W/ ARMV5TE) ARCH

ty of network protocols.

**stand by Wi-Fi STA , AP , Direct , Repeater mode**

**stand by SGI , Green-Field Preamble and A-MPDU**

**stand by WPA , WPA2 with WAPI Security Protocol**

**double I2C**

**stand by 802.11e as well as WMM-PS protocol**

anel ADC An input voltage

**ARM968E-S MCU The highest frequency 120 MHz**

e Support

**Chip FLASH , Support for transparent download**

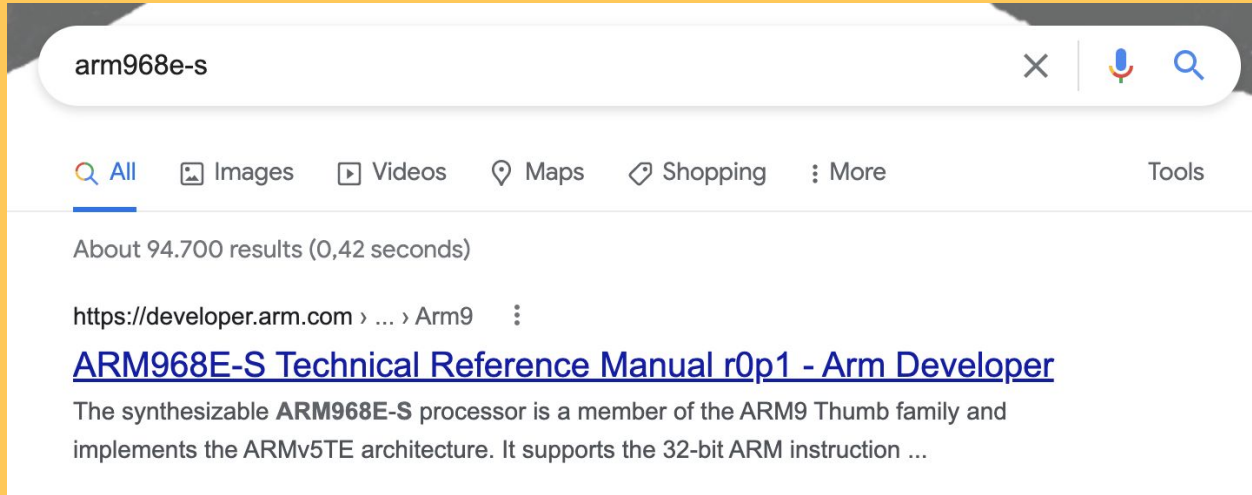
**Chip 256 Kbyte data RAM 50 MHz SDIO Interface and**

F switches, using QFN Package,

**single SPI interface**

# BK7231 ARCHITECTURE

## 968E-S (ARM9 W/ ARMV5TE) ARCH





# BK7231 ARCHITECTURE

## 968E-S (ARM9 W/ ARMV5TE) ARCH

### Contents

## ARM968E-S Technical Reference Manual

#### Preface

About this manual .....	xii
Feedback .....	xvi

#### Chapter 1

#### Introduction

1.1	About the ARM968E-S processor .....	1-2
1.2	TCM access .....	1-5
1.3	Debug interface configurations .....	1-6



bk7231t



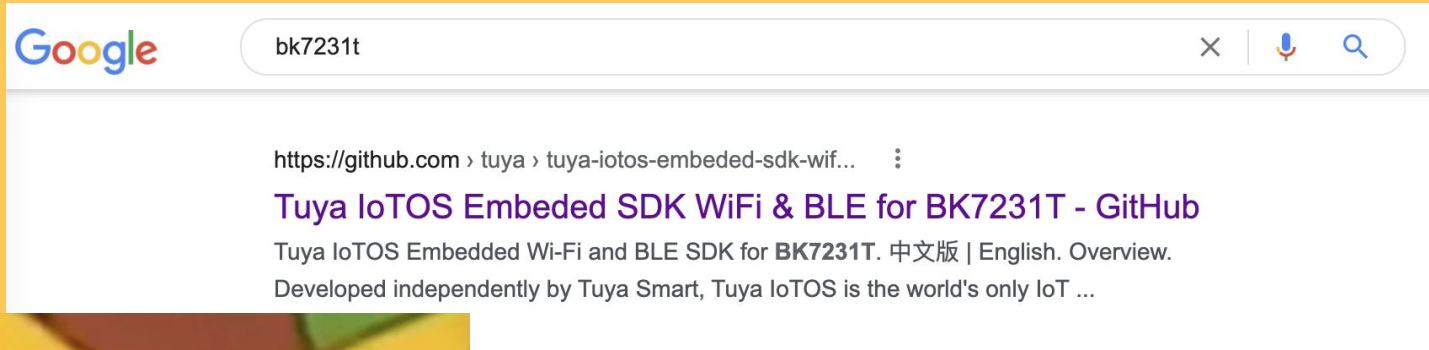
<https://github.com> › [tuya](#) › [tuya-iotos-embedded-sdk-wif...](#) ⋮

## Tuya IoTOS Embeded SDK WiFi & BLE for BK7231T - GitHub

Tuya IoTOS Embedded Wi-Fi and BLE SDK for **BK7231T**. 中文版 | English. Overview.

Developed independently by Tuya Smart, Tuya IoTOS is the world's only IoT ...

# OPEN SOURCE!



A screenshot of a Google search interface. The search bar contains the text "bk7231t". Below the search bar, the search results are displayed. The first result is a link to a GitHub repository: "https://github.com > tuya > tuya-iotos-embedded-sdk-wif...". The title of the repository is "Tuya IoTOS Embeded SDK WiFi & BLE for BK7231T - GitHub". Below the title, there is a description: "Tuya IoTOS Embedded Wi-Fi and BLE SDK for **BK7231T**. 中文版 | English. Overview. Developed independently by Tuya Smart, Tuya IoTOS is the world's only IoT ...".



# OPEN SOURCE!

## WITH SOME BLOBS

The screenshot shows the GitHub interface for the repository 'tuya / tuya-iotos-embedded-sdk-wifi-ble-bk7231t'. The repository is public and has a 'Code' button highlighted. The navigation bar includes links for Code, Issues (2), Pull requests, Actions, Projects, Wiki, Security, and Insights. Below the navigation bar, there are buttons for 'Go to file', 'Add file', and 'Code'. The main content area displays a list of files and folders, including 'apps', 'platforms/bk7231t', 'sdk', '.gitignore', 'README.md', 'README\_zh.md', and 'build\_app.sh'. Each item shows the commit message and the time since the last commit.

tuya / tuya-iotos-embedded-sdk-wifi-ble-bk7231t Public

<> Code Issues 2 Pull requests Actions Projects Wiki Security Insights

master 1 branch 1 tag Go to file Add file Code

chenyisong update package.exe	4c04a6c on 23 Sep 2021	9 commits
apps	update template_demo PRODECT_ID	13 months ago
platforms/bk7231t	update package.exe	6 months ago
sdk	tuya iotos embedded wifi&ble sdk for bk7231t version 1.0.2	14 months ago
.gitignore	tuya iotos embedded wifi&ble sdk for bk7231t version 1.0.2	14 months ago
README.md	Update README.md	13 months ago
README_zh.md	fix README.md link address	14 months ago
build_app.sh	tuya iotos embedded wifi&ble sdk for bk7231t version 1.0.2	14 months ago

☰ README.md

Tuya IoTOS Embedded Wi-Fi and BLE SDK for BK7231T

[中文版](#) | [English](#)

## Overview

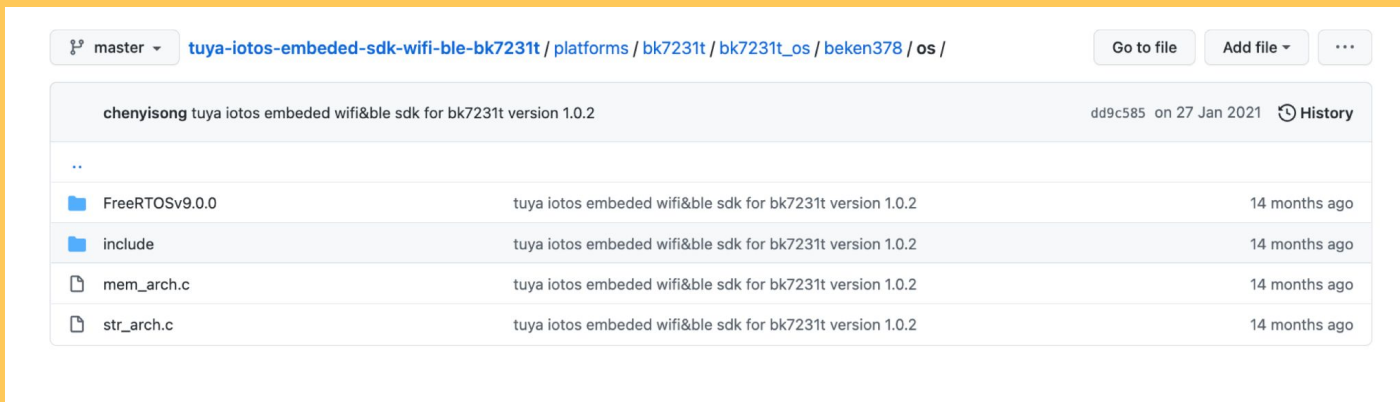
---

Developed independently by Tuya Smart, Tuya **IoTOS** is the world's only IoT operating system covering all levels of IoT sensing, interruption, network, platform, and application. Benefiting from Tuya Smart's accumulation in the IoT industry, TuyaIoTOS provides solutions for a full range of products from product design, R&D, to post-operation.

Tuya IoTOS embedded SDK is an important part of Tuya IoTOS. By virtue of dedicated design, it provides customers with unified APIs, rich SDKs, and DIY functions, enhancing the integrality of the IoT industry. It can be applied to industrial IoT, vehicle networking, security monitoring, outing, and smart home development.

# IOTOS

## ACTUALLY FREERTOS



The screenshot shows a GitHub repository interface for the path `tuya-iotos-embedded-sdk-wifi-ble-bk7231t / platforms / bk7231t / bk7231t_os / beken378 / os /`. The repository is named `chenyisong tuya iotos embeded wifi&ble sdk for bk7231t version 1.0.2` and was last updated on 27 Jan 2021. The file browser displays the following directory structure:

Item	Description	Last Modified
..		
FreeRTOSv9.0.0	tuya iotos embeded wifi&ble sdk for bk7231t version 1.0.2	14 months ago
include	tuya iotos embeded wifi&ble sdk for bk7231t version 1.0.2	14 months ago
mem_arch.c	tuya iotos embeded wifi&ble sdk for bk7231t version 1.0.2	14 months ago
str_arch.c	tuya iotos embeded wifi&ble sdk for bk7231t version 1.0.2	14 months ago

# A NEW CHIP, A NEW BUG

```
struct LAN_AP_NW_CFG_S {
    // ...
    char ap_cfg_token[64];
    int fd;
    short log_ack_timer;
    short send_log_mid;
    int (*finish_cb)(PTR_SSID_PASSWORD_TOKEN, int);
    SSID_PASSWORD_TOKEN spt;
    // ...
};

struct LAN_AP_NW_CFG_S lan_ap_nw_cfg;

//...

void __udp_ap_v3_task()
{
    // ...
    cJSON *json_object = parse_json_payload();
    cJSON *ssid = cJSON_GetObjectItem("ssid", json_object);
    cJSON *password = cJSON_GetObjectItem("passwd", json_object);
    cJSON *token = cJSON_GetObjectItem("token", json_object);
    // ...
    strncpy(lan_ap_nw_cfg->spt.ssid, ssid->valuestring, 32);
    strncpy(lan_ap_nw_cfg->spt.passwd, password->valuestring, 64);

    int token_length = strlen(token);
    memcpy(lan_ap_nw_cfg.ap_cfg_token, token, token_length);

    int result = lan_ap_nw_cfg->finish_cb(lan_ap_nw_cfg->spt, 0x10002);
    cJSON_Delete(json_object);
    // ...
}
```

# A NEW CHIP, A NEW BUG

```
struct LAN_AP_NW_CFG_S {
    // ...
    char ap_cfg_token[64];
    int fd;
    short log_ack_timer;
    short send_log_mid;
    int (*finish_cb)(PTR_SSID_PASSWORD_TOKEN, int);
    SSID_PASSWORD_TOKEN spt;
    // ...
};

struct LAN_AP_NW_CFG_S lan_ap_nw_cfg;

//...

void __udp_ap_v3_task()
{
    // ...
    cJSON *json_object = parse_json_payload();
    cJSON *ssid = cJSON_GetObjectItem("ssid", json_object);
    cJSON *password = cJSON_GetObjectItem("passwd", json_object);
    cJSON *token = cJSON_GetObjectItem("token", json_object);
    // ...
    strncpy(lan_ap_nw_cfg->spt.ssid, ssid->valuestring, 32);
    strncpy(lan_ap_nw_cfg->spt.passwd, password->valuestring, 64);

    int token_length = strlen(token);
    memcpy(lan_ap_nw_cfg.ap_cfg_token, token, token_length);

    int result = lan_ap_nw_cfg->finish_cb(lan_ap_nw_cfg->spt, 0x10002);
    cJSON_Delete(json_object);
    // ...
}
```

```
{
    "ssid": "AP SSID",
    "passwd": "AP passphrase",
    "token": "AP configuration token"
}
```



# A NEW CHIP, A NEW BUG

```
struct LAN_AP_NW_CFG_S {
    // ...
    char ap_cfg_token[64];
    int fd;
    short log_ack_timer;
    short send_log_mid;
    int (*finish_cb)(PTR_SSID_PASSWORD_TOKEN, int);
    SSID_PASSWORD_TOKEN spt;
    // ...
};

struct LAN_AP_NW_CFG_S lan_ap_nw_cfg;

//...

void __udp_ap_v3_task()
{
    // ...
    cJSON *json_object = parse_json_payload();
    cJSON *ssid = cJSON_GetObjectItem("ssid", json_object);
    cJSON *password = cJSON_GetObjectItem("passwd", json_object);
    cJSON *token = cJSON_GetObjectItem("token", json_object);
    // ...
    strncpy(lan_ap_nw_cfg->spt.ssid, ssid->valuestring, 32);
    strncpy(lan_ap_nw_cfg->spt.passwd, password->valuestring, 64);

    int token_length = strlen(token);
    memcpy(lan_ap_nw_cfg.ap_cfg_token, token, token_length);

    int result = lan_ap_nw_cfg->finish_cb(lan_ap_nw_cfg->spt, 0x10002);
    cJSON_Delete(json_object);
    // ...
}
```

```
{
    "ssid": "AP SSID",
    "passwd": "AP passphrase",
    "token": "AP configuration token"
}
```

# A NEW CHIP, A NEW BUG

```
struct LAN_AP_NW_CFG_S {
    // ...
    char ap_cfg_token[64];
    int fd;
    short log_ack_timer;
    short send_log_mid;
    int (*finish_cb)(PTR_SSID_PASSWORD_TOKEN, int);
    SSID_PASSWORD_TOKEN spt;
    // ...
};

struct LAN_AP_NW_CFG_S lan_ap_nw_cfg;

//...

void __udp_ap_v3_task()
{
    // ...
    cJSON *json_object = parse_json_payload();
    cJSON *ssid = cJSON_GetObjectItem("ssid", json_object);
    cJSON *password = cJSON_GetObjectItem("passwd", json_object);
    cJSON *token = cJSON_GetObjectItem("token", json_object);
    // ...
    strncpy(lan_ap_nw_cfg->spt.ssid, ssid->valuestring, 32);
    strncpy(lan_ap_nw_cfg->spt.passwd, password->valuestring, 64);

    int token_length = strlen(token);
    memcpy(lan_ap_nw_cfg.ap_cfg_token, token, token_length);

    int result = lan_ap_nw_cfg->finish_cb(lan_ap_nw_cfg->spt, 0x10002);
    cJSON_Delete(json_object);
    // ...
}
```

```
{
    "ssid": "AP SSID",
    "passwd": "AP passphrase",
    "token": "AP configuration token"
}
```

# A NEW CHIP, A NEW BUG

```
struct LAN_AP_NW_CFG_S {
    // ...
    char ap_cfg_token[64];
    int fd;
    short log_ack_timer;
    short send_log_mid;
    int (*finish_cb)(PTR_SSID_PASSWORD_TOKEN, int);
    SSID_PASSWORD_TOKEN spt;
    // ...
};

struct LAN_AP_NW_CFG_S lan_ap_nw_cfg;

//...

void __udp_ap_v3_task()
{
    // ...
    cJSON *json_object = parse_json_payload();
    cJSON *ssid = cJSON_GetObjectItem("ssid", json_object);
    cJSON *password = cJSON_GetObjectItem("passwd", json_object);
    cJSON *token = cJSON_GetObjectItem("token", json_object);
    // ...
    strncpy(lan_ap_nw_cfg->spt.ssid, ssid->valuestring, 32);
    strncpy(lan_ap_nw_cfg->spt.passwd, password->valuestring, 64);

    int token_length = strlen(token);
    memcpy(lan_ap_nw_cfg.ap_cfg_token, token, token_length);

    int result = lan_ap_nw_cfg->finish_cb(lan_ap_nw_cfg->spt, 0x10002);
    cJSON_Delete(json_object);
    // ...
}
```

```
{
    "ssid": "AP SSID",
    "passwd": "AP passphrase",
    "token": "AP configuration token"
}
```

# DUMPING FIRMWARE

# DUMPING FIRMWARE

## SERIAL PROTOCOL

- FLASH READING/WRITING
- DEVICE METADATA
- SPECIFICATION IS .. UNKNOWN

# DUMPING FIRMWARE

## SERIAL PROTOCOL

- FLASH READING/WRITING
- DEVICE METADATA
- SPECIFICATION IS .. UNKNOWN  
SORT OF UNKNOWN

Module pin	Serial pin
RXD1	TX
TXD1	RX
VCC	VCC3.3V
GND	GND

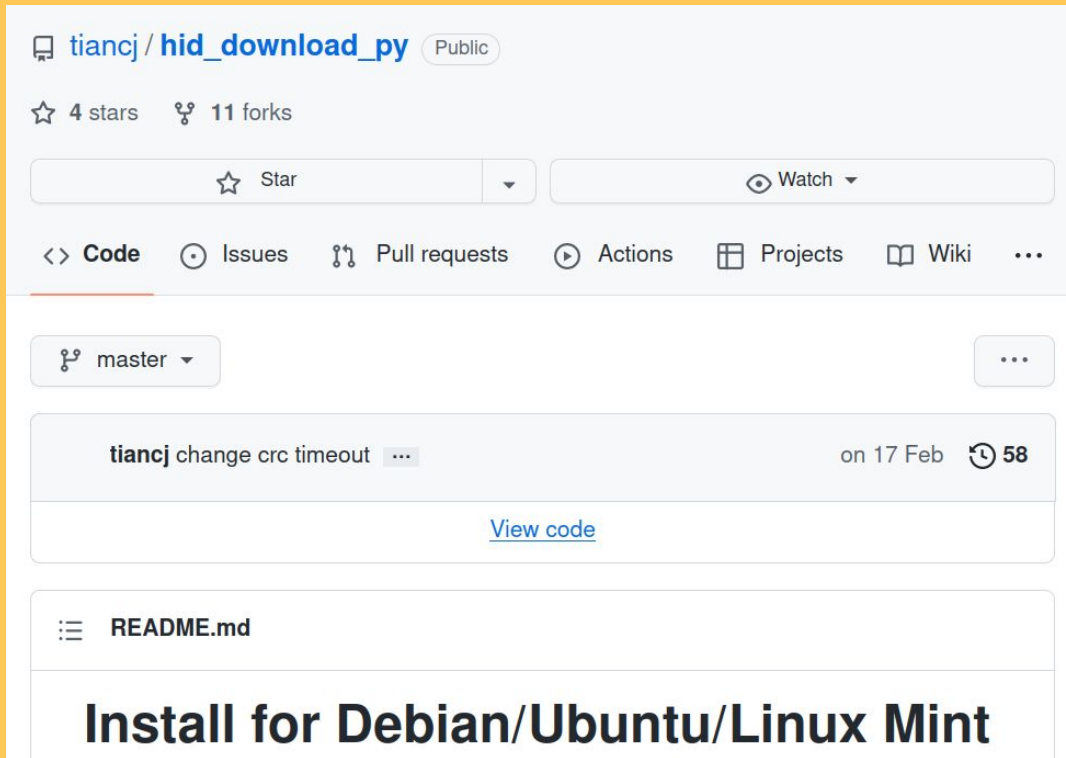
## Prepare software [↗](#)

Download and open the [BK7231T chip flashing tool](#). The following figure shows the window of the chip flashing tool. The following table describes the required parameters boxed in red in the figure.

Parameter	Description
Flashing Target	The chip platform is BK7231T. Therefore, <b>BK7231 is used.</b>
Start address	Set the value to <code>0x00011000</code> in most cases.
Operation length	Select <code>0x001EF000 (0x00200000-0x00011000)</code>
Baud rate	Select <code>921600</code>

# DUMPING FIRMWARE

MOAR OPEN SOURCE!



The screenshot shows the GitHub interface for the repository 'tiancj/hid\_download\_py'. At the top, it indicates the repository is 'Public'. Below this, it shows '4 stars' and '11 forks'. There are two input fields: one for 'Star' and one for 'Watch'. A navigation bar contains links for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', and a menu icon. Below the navigation bar, there is a dropdown menu for branches, currently set to 'master'. A commit history entry is visible, showing a commit by 'tiancj' titled 'change crc timeout' on '17 Feb' with '58' comments. A 'View code' link is provided for this commit. Below the commit history, there is a section for 'README.md' with a hamburger menu icon. The main content area displays the heading 'Install for Debian/Ubuntu/Linux Mint'.

tiancj / [hid\\_download\\_py](#) Public

☆ 4 stars 🍴 11 forks

☆ Star Watch

<> Code Issues Pull requests Actions Projects Wiki ...

🍴 master ...

tiancj change crc timeout ... on 17 Feb 🕒 58

[View code](#)

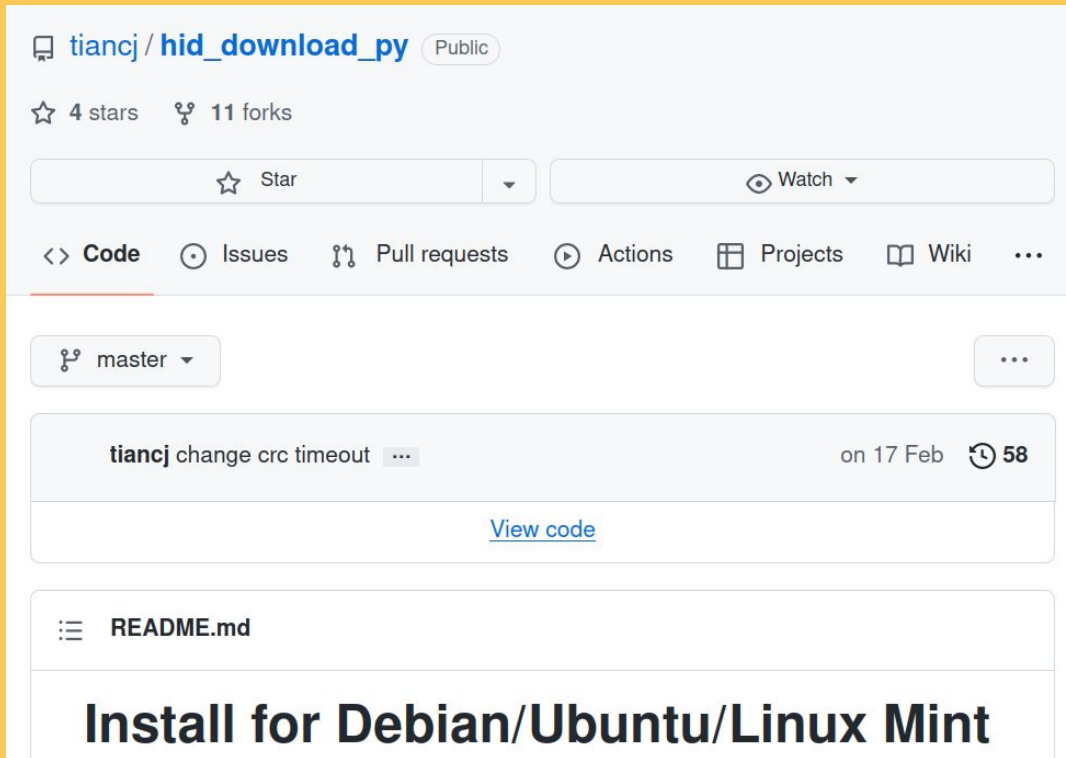
☰ README.md

## Install for Debian/Ubuntu/Linux Mint

# DUMPING FIRMWARE

MOAR OPEN SOURCE!

... BUT IT ISN'T RELIABLE.



The screenshot shows a GitHub repository page for 'tiancj / hid\_download\_py'. The repository is public and has 4 stars and 11 forks. The page includes navigation links for Code, Issues, Pull requests, Actions, Projects, Wiki, and a menu icon. A commit by 'tiancj' titled 'change crc timeout' is visible, dated 'on 17 Feb' with 58 comments. Below the commit, there is a 'View code' link and a file named 'README.md'. The bottom of the page features the heading 'Install for Debian/Ubuntu/Linux Mint'.

tiancj / [hid\\_download\\_py](#) Public

☆ 4 stars 🍴 11 forks

☆ Star Watch

<> Code Issues Pull requests Actions Projects Wiki ...

🍴 master ...

tiancj change crc timeout ... on 17 Feb 🕒 58

[View code](#)

☰ README.md

## Install for Debian/Ubuntu/Linux Mint



# DUMPING FIRMWARE

## REVERSING THE BOOTLOADER AND COMPARING WITH THE OPEN SOURCE TOOL

```
case 8:
    zero_var = *(ushort*)(maybe_parse_state + 2);
    cmd_type_var = *request_bytes;
    one_var_2byte = zero_var + 1 & 0xfffffff;
    *(short*)(maybe_parse_state + 2) = (short)one_var_2byte;
    puVar3[zero_var] = cmd_type_var;
    cmd_type_again = *puVar3;
    if ((cmd_type_again == '\x09') && (one_var_2byte == *(ushort*)(maybe_parse_state + 8))) {
        /* if command type == 0x9
           and command length lsb == 0x1

           This seems to be flash read 4k */
        addr_for_cmd = *(uint*)(puVar3 + 1);
        *(uint*)(maybe_parse_state + 0x1c) = addr_for_cmd;
        /* if addr < 0x10000 */
        if (addr_for_cmd < 0x10000) {
            puVar3[5] = (char)*(ushort*)(maybe_parse_state + 8) + -5;
            maybe_build_cmd_response(9,6,7,puVar3 + 1);
        }
    }
}
```

```
242 def BuildCmd_FlashRead4K(addr: int):
243     length=1+(4+0)
244     buf = bytearray(4096)
245     buf[0]=0x01
246     buf[1]=0xe0
247     buf[2]=0xfc
248     buf[3]=0xff
249     buf[4]=0xf4
250     buf[5]=(length&0xff)
251     buf[6]=((length>>8)&0xff)
252     buf[7]=CMD_FlashRead4K
```

# DUMPING FIRMWARE

## CREATING A NEW TOOL FOR THE BOOTLOADER SERIAL PROTOCOL

☰ README.md ✎

---

# bk7231tools

---

This is a collection of tools to interact with and analyze artifacts for BK7231 MCUs.

## Contributors

---




- [Kuba Szczodrzyński - @kuba2k2](#)

### Packages

No packages published  
[Publish your first package](#)

---

### Contributors 3

-  **notkmhn** Khaled Nassar
-  **tjclement** Tom Clement
-  **kuba2k2** Kuba Szczodrzyński

# **ANALYZING FIRMWARE DUMPS**

# ANALYZING FIRMWARE DUMPS

Flash layout contains two code partitions

- Bootloader
- User app

Other configuration partitions too, defined by user app

# ANALYZING FIRMWARE DUMPS

Flash layout contains two code partitions

- Bootloader
- User app

Other configuration partitions too, defined by user app



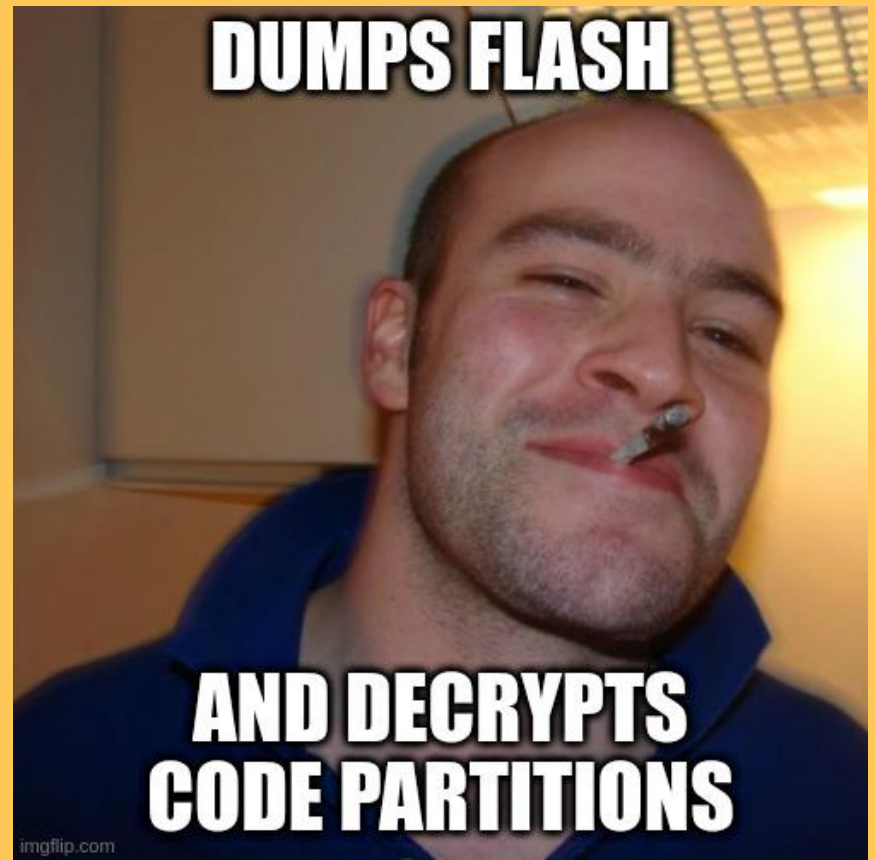
AKA OBFUSCATION

## ANALYZING FIRMWARE DUMPS

Flash layout contains two code partitions

- Bootloader
- User app

Other configuration partitions too, defined by user app



**BK7231TOOLS TO THE RESCUE**

**BUG TO JAILBREAK**

# BUG TO JAILBREAK

```
struct LAN_AP_NW_CFG_S {
    // ...
    char ap_cfg_token[64];
    int fd;
    short log_ack_timer;
    short send_log_mid;
    int (*finish_cb)(PTR_SSID_PASSWORD_TOKEN, int); ←
    SSID_PASSWORD_TOKEN spt;
    // ...
};

struct LAN_AP_NW_CFG_S lan_ap_nw_cfg;

//...

void __udp_ap_v3_task()
{
    // ...
    cJSON *json_object = parse_json_payload();
    cJSON *ssid = cJSON_GetObjectItem("ssid", json_object);
    cJSON *password = cJSON_GetObjectItem("passwd", json_object);
    cJSON *token = cJSON_GetObjectItem("token", json_object);
    // ...
    strncpy(lan_ap_nw_cfg->spt.ssid, ssid->valuestring, 32);
    strncpy(lan_ap_nw_cfg->spt.passwd, password->valuestring, 64);

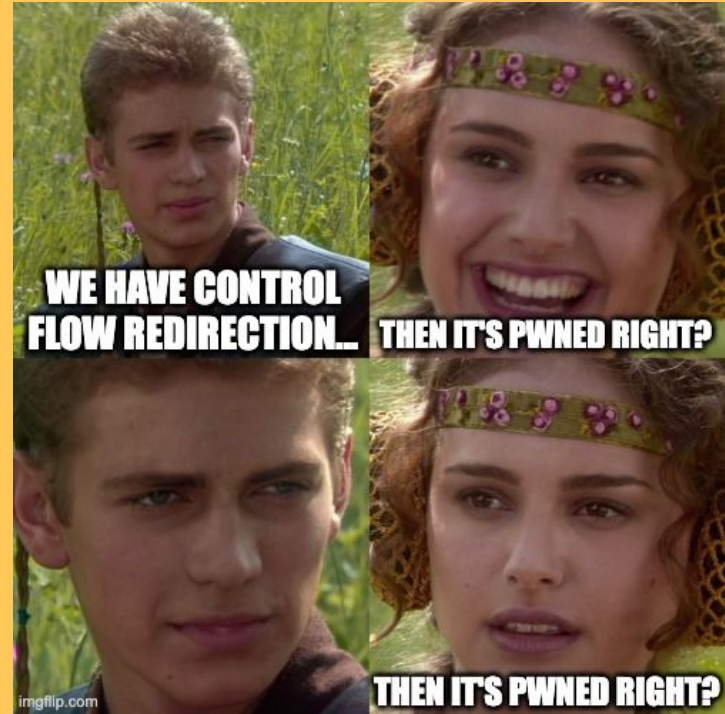
    int token_length = strlen(token);
    memcpy(lan_ap_nw_cfg.ap_cfg_token, token, token_length);

    int result = lan_ap_nw_cfg->finish_cb(lan_ap_nw_cfg->spt, 0x10002); ←
    cJSON_Delete(json_object);
    // ...
}
```

```
{
    "ssid": "AP SSID",
    "passwd": "AP passphrase",
    "token": "AP configuration token"
}
```



**DONE! OR NOT..?**



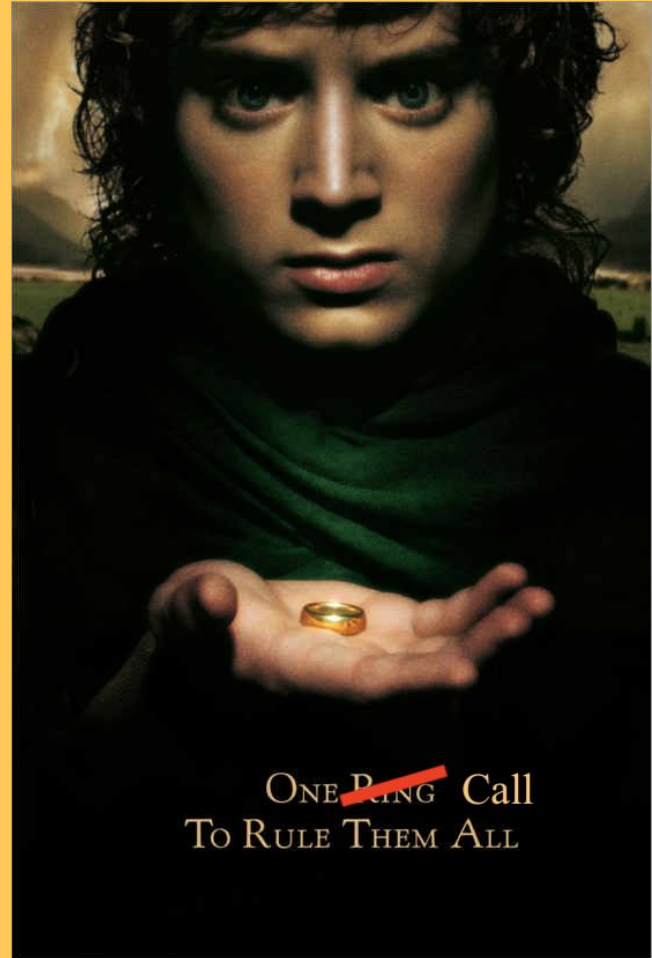
# OVERWRITING THE PSK

```
void __tmm_gw_pskkey_get_cb(s_tm_msg *tm_msg)
{
    ...
    OVar1 = http_pskkey_get(&result);
    if (OVar1 == 0) {
        ...
    }
    else {
        ptVar2 = ty_cJSON_GetObjectItem(result, "pskKey");
        __src = ptVar2->valuelstring;
        PrintLog(...);
        strncpy(gw_cntl.gw_base.psk_key, __src, 0x29);
        OVar1 = wd_gw_base_if_write(&gw_cntl.gw_base);
        if (OVar1 == 0) {
            tuya_tls_register_constant
                (gw_cntl.gw_base.uuid, gw_cntl.gw_base.auth_key, gw_cntl.gw_base.psk_key);
        }
        else {
            PrintLog(...);
        }
        ty_cJSON_Delete(result);
    }
    ...
}
```

# REASSESSING WIN CONDITIONS

- Get the PSK
  - Overwrite
  - Leak
- ~~Downgrade to vulnerable protocol version~~
- Get code execution on the device itself
- Overwrite security keys

# OVERWRITING KEYS



# GADGET HUNTING

- Function that sets the security keys to jump into
- A "fixup" gadget to set up the registers as needed

# GADGET HUNTING

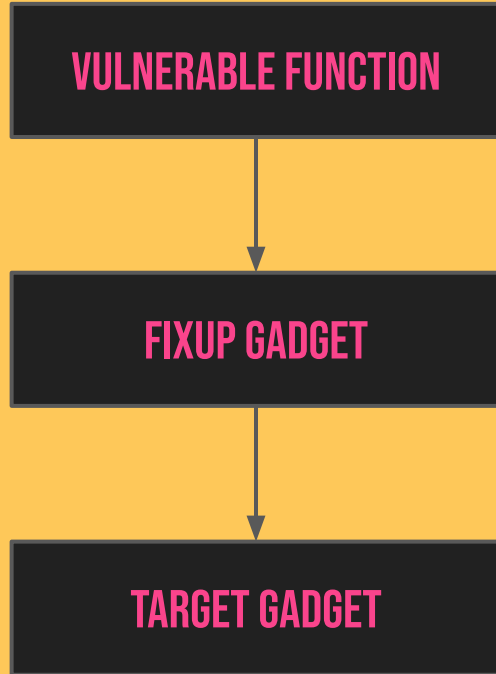
## TARGET GADGET

```
void __mf_cmd_process(MF_PRO_HEAD_S *hd) {  
    ...  
    ptVar17 = ty_cJSON_GetObjectItem(ptVar9, "uuid");  
    strcpy((char *)pGVar18, ptVar17->valuestring);  
    ptVar17 = ty_cJSON_GetObjectItem(ptVar9, "pskKey");  
    strcpy(pGVar18->psk_key, ptVar17->valuestring);  
    ptVar17 = ty_cJSON_GetObjectItem(ptVar9, "auzkey");  
    strcpy(pGVar18->auth_key, ptVar17->valuestring);  
    ...  
    OVar10 = wd_gw_base_if_write(pGVar18);  
    ...  
}
```

## TRAMPOLINE GADGET

```
adds r0, r7, #0  
ldr r1, [sp, #8]  
ldr r3, [r5, #0x20]  
blx r3
```

# PUTTING EVERYTHING TOGETHER



# RELEASING INTO THE WILD

**DO YOU HAVE A WORKING  
JAILBREAKING TOOL AVAILABLE?**



**I THOUGHT WE  
COULD JUST SHARE OUR POC**





# RELEASING INTO THE WILD

## TUYA-CLOUDCUTTER



tuya-cloudcutter / **tuya-cloudcutter** Public

A tool that disconnects Tuya devices from the cloud, allowing them to run completely locally.

MIT license

☆ 68 stars    🍴 11 forks

☆ Star    🔔 Notifications

<> **Code**    ⓘ Issues 51    🔗 Pull requests 1    ▶ Actions    📁 Projects    📖 Wiki    ⋮

🔗 main ▾    [Go to file](#)

**Khaled Nassar** Merge pull request #110 from Cossid/Euarne-BR30-RG... ... 27 days ago 🔄 176

[View code](#)

**DEMO TIME**

## IN CLOSING

- Found a vulnerability affecting almost all BK7231 devices to date
- Tuya was really cool about it -> we sent bug bounty to charity
- Embedded security is catching up -> still interesting target

# GETTING IN TOUCH

## KHALED NASSAR

 notkmhn



 @notkmhn

## TOM CLEMENT

 tjclement

 @Tom\_Clement

# ACKNOWLEDGMENTS

- ius - ESP8266 JTAG debugging
- blasty ( @bl4sty) - ESP8266 vulnerability analysis support
- Jilles Groenendijk ( @jilles\_com) - BK7231 firmware gathering and support
- V-TRUST for disclosure tips