argXtract: Deriving IoT Security Configurations via Automated Static Analysis of Stripped ARM Cortex-M Binaries

Pallavi Sivakumaran, Jorge Blasco

EPSRC
Engineering and Physical Sciences Research Council

Centre for Doctoral Training in Cyber Security
Information Security Group
Royal Holloway University of London

The cyber-attack that brought d
was caused by a new weapon called the Mirai botnet and wa

DDoS attack that disrupted internet was largest of its kind in history, experts say

and remained under sustained assault for most of the day, bringing d
sites including Twitter, the Guardian, Netflix, Reddit, CNN and many
and the US.

distributed denial of service (DDoS) at
special malware, know
traffic unti

Wi-Fi baby heart monitor may have the worst IoT security of

ast year when suddenly the boar
le couldn't control the board a
mechanical defect, so being a compu
new to other scenarios: could he have been hacked?

It didn't take long to determine that Bluetooth noise in the neighborhood was the likely
culprit. The intersection, near Federation Square, was notorious for being saturated with
radio frequency noise. Healey was control
handheld remote that sent
drive commands to the board via Bluetoo
concluded, a flood of Bluetooth traffic fr
remote's connection to the board.

hat was wrong. There was
naturally

Hacking risk leads to recall of 500,000 pacemakers due to patient death fears

Drug Administration (FDA) due to fears that their lax cybersecurity could be
hacked to run the batteries down or even alter the patient's heartbeat.

The recall won't see the pacemakers removed, which would be an invasive
d dangerous medical procedure for the 465,000 people who have them
nplanted: instead, the manufacturer has issued a firmware update which
ill be applied by medical staff to patch the security holes.

aker, all made by healthtech firm Abbott and sold under
brand, are affected by the recall. They are all radio-
cardiac pacemakers, typically fitted to patients with
eats, as well as those recovering from heart failure.

s of unauthorised access to any patient's
to Abbot. The FDA says that the vulne
o access a device using commer
he hackers could then deli
tration of inappropr
the death of a
curity said
te with the
urity attack

esident of medical devi
nstantly vigilant against unauth
ess, which is why we're working with ot
tor to ensure we're proactively addressing comm
advance the security of devices and systems."

s the second round of updates for the heart implants that Abbott
unced since buying medical device maker St. Jude Medical earlier

he weaknesses were discovered by MedSec, a cybersecu
specialises in

Smart camera and baby monitor warning given by UK's cyber-defender

Smart cameras and baby monitors can be watched by criminals over the internet by default, security chiefs warn.

The National Cyber Security Centre (NCSC) is advising people to tweak the settings after buying them.

Easy-to-guess default passwords might let a hacker secretly observe a home through connected devices, it said.

The NCSC's technical director, Dr Ian Levy, warned while the devices were "fantastic innovations", they were vulnerable to cyber-attackers.

There are many examples of devices being accessed without permission.

In one, **the attacker spoke to a young girl**, pretending to be Father Christmas.

In another, a couple from Leeds had been watched thousands of times online
t their knowledge.

searchers **easily breached an adult toy that had a camera**

Kids' Smartwatches Are a Security Nightmare Despite Years of Warnings

CONNECTING EVERY POSSIBLE device in our lives to the internet has always represented a security risk. But that risk is far more pronounced when it involves a smartwatch strapped to your child's wrist. Now, even after years of warnings about the security failings of many of those devices, one group of researchers has shown that several remain appallingly easy hackers to abuse.

In a paper published late last month, researchers at the Münster Univ
Sciences in Germany detailed their testing of the security of s
marketed for kids. They're designed to send and recei
parents track their child's location from a smartphone
hackers could abuse those features to track a target chi
ive out of the six br
ulnerabilities, allo
ome from their pa
en to record aud
earchers share
eral of the bug
Münster stud
rtwatches ha
umer protec
mission eve
ted expos

Several of t

who wo
ey could be h the St
them com

ix types of
aker, all made by healthtech firm Abbott and sold under
brand, are affected by the recall. They are all radio-
cardiac pacemakers, typically fitted to patients with
eats, as well as those recovering from heart failure.

Ring Plagued by Security Issues, Flood of Hacks

Serious security holes in the Ring smart doorbell have been uncovered, according to a new
investigation. For instance, Ring owners aren't notified of suspicious login alerts when
ices are accessed on various IP addresses — and there are seemingly no limitations for
t login attempts.

based on Motherboard's security tests on the Amazon-owned
ome on the heels of several privacy and security incidents relating to
cludes several disturbing stories emerging over the past week of
and stalking to strangers through them.

security tests, Ring said that many customers were
ther services, allowing bad actors to gain

Thousands of webcams vulnerable to attack

More than 15,000 webcams in homes and offices can be accessed remotely by anyone if users implement
manipulated over just an internet connection.

Many security and conferencing cameras can be accessed remotely by anyone if users implement
additional security measures post-installation, according to findings by Avishai Ef
with Wizcase. In other cases, these cameras are set with predicta
credentials.

Webcams susceptible to this i
and WebCamXP 5 software, am

Many may assume that only devices
that connect other d
way

Hackers Can Seize Control of Electric Skateboards and Toss Riders

Root Cause?

CVE-2015-2880: TRENDnet WiFi Baby Cam TV-IP743SIC has a password of admin for the backdoor root account

CVE-2017-17436: An issue was discovered in the software on Vaultek Gun Safe VT20i products. There is no encryption of the session between the Android application and the safe

CVE-2021-33220: An issue was discovered in CommScope Ruckus IoT Controller 1.7.1.0 and earlier. Hard-coded API Keys exist

>> configuration issues

Source of Configuration Data?

- Devices
- Mobile Apps
- Web Portals
- Firmware

Problems

- Firmware analysis is complex
- Firmware analysis for hub/gateway devices may try to identify familiar filesystem structures or standard functions, due to the use of Linux/VxWorks-based OSs
- *Peripheral* firmware analysis can be *very* complex
  - No/non-standard OS
  - The security-relevant configuration calls may be made from an application to an underlying technology stack, where the stack is often shipped separately from the application firmware
  - Tend to be available only as stripped binaries (no ELF headers/symbols/section information)
  - Increasingly featuring Cortex-M processors (Thumb instructions, often inline data)

SOURCE

```
uint8_t passkey[] = "123456";
ble_opt_t ble_opt;
ble_opt.gap_opt.passkey.p_passkey = &passkey[0];
err_code = sd_ble_opt_set(BLE_GAP_OPT_PASSKEY, &ble_opt);
```

DISASSEMBLY OF UNSTRIPPED BINARY

```
Instruction address   Hex representation   Opcode   Operands

1eaba:   4ab8       ldr  r2, [pc, #736] ;(1ed9c)
1eabc:   ab06       add  r3, sp, #24
1eabe:   6811       ldr  r1, [r2, #0]
1eac0:   2022       movs r0, #34
1eac2:   9106       str  r1, [sp, #24]
1eac4:   8891       ldrh r1, [r2, #4]
1eac6:   8099       strh r1, [r3, #4]
1eac8:   7992       ldrb r2, [r2, #6]
1eaca:   a908       add  r1, sp, #32
1eacc:   719a       strb r2, [r3, #6]
1eace:   9308       str  r3, [sp, #32]
1ead0:   f7fffe3a   bl   1e748 <sd_ble_opt_set>

1ed9c:   00021f14   .word  0x00021f14



21f0c:   … 3231 3433 3635 0000  (%. .123456..
```

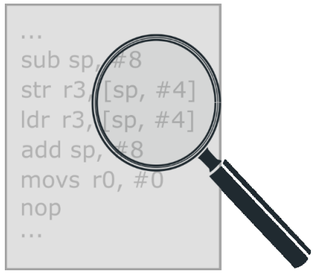
DISASSEMBLY OF STRIPPED BINARY

```
Instruction address   Hex representation   Opcode   Operands

0x3aba:   b84a      ldr   r2, [pc, #0x2e0]
0x3abc:   06ab      add   r3, sp, #0x18
0x3abe:   1168      ldr   r1, [r2]
0x3ac0:   2220      movs  r0, #0x22
0x3ac2:   0691      str   r1, [sp, #0x18]
0x3ac4:   9188      ldrh  r1, [r2, #4]
0x3ac6:   9980      strh  r1, [r3, #4]
0x3ac8:   9279      ldrb  r2, [r2, #6]
0x3aca:   08a9      add   r1, sp, #0x20
0x3acc:   9a71      strb  r2, [r3, #6]
0x3ace:   0893      str   r3, [sp, #0x20]
0x3ad0:   fff73afe  bl    #0x3748

0x3d9c:   141f   subs r4, r2, #4
0x3d9e:   0200   movs r2, r0

0x6f14:   3132   adds r2, #0x31
0x6f16:   3334   adds r4, #0x33
0x6f18:   3536   adds r6, #0x35
```

no function information

data interpreted as code

incorrect instruction addresses



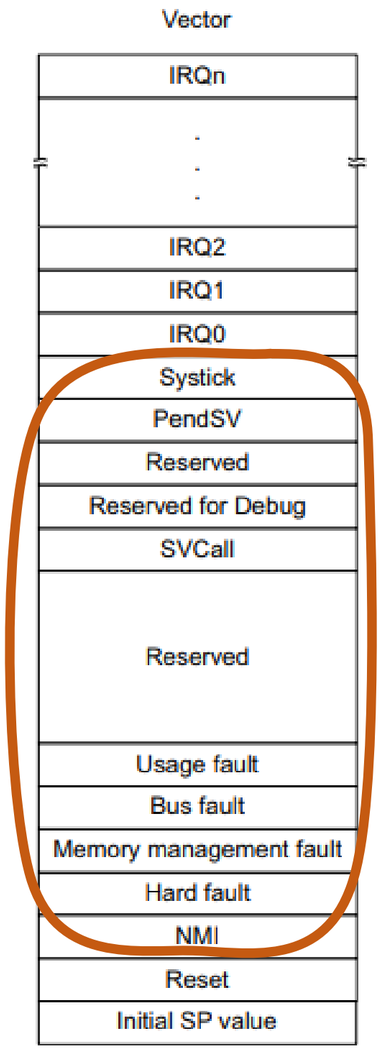
argXtract

1. Identify app code base using Vector Table and self-targeting branches.
2. Identify inline data using Reset Handler, LDR, table branches.
3. Estimate function blocks using Vector Table, BL, exit bypasses.
4. Pinpoint instructions of interest as SVCalls or function calls.
5. Generate trace chains and trace through assembly keeping track of registers/RAM.
6. Process register/memory objects according to Argument Definition Objects.

>> Ultimately extract arguments to (security-relevant) functions of interest

https://github.com/projectbtle/argXtract

1. Identify app code base



nrf51422_xxac_s130.bin

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00000000	00	80	00	20	31	1A	02	00	71	1A	02	00	73	1A	02	00
00000010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000020	00	00	00	00	00	00	00	00	00	00	00	00	75	1A	02	00
00000030	00	00	00	00	00	00	00	00	77	1A	02	00	79	1A	02	00
00000040	7B	1A	02	00	7B	1A	02	00	61	C5	01	00	7B	1A	02	00
00000050	7B	1A	02	00	00	00	00	00	45	C0	01	00	7B	1A	02	00
00000060	7B	1A	02	00	7B	1A	02	00	7B	1A	02	00	7B	1A	02	00
00000070	7B	1A	02	00	7B	1A	02	00	7B	1A	02	00	7B	1A	02	00
00000080	7B	1A	02	00	B9	B4	01	00	7B	1A	02	00	7B	1A	02	00
00000090	95	B5	01	00	7B	1A	02	00	A1	19	02	00	7B	1A	02	00

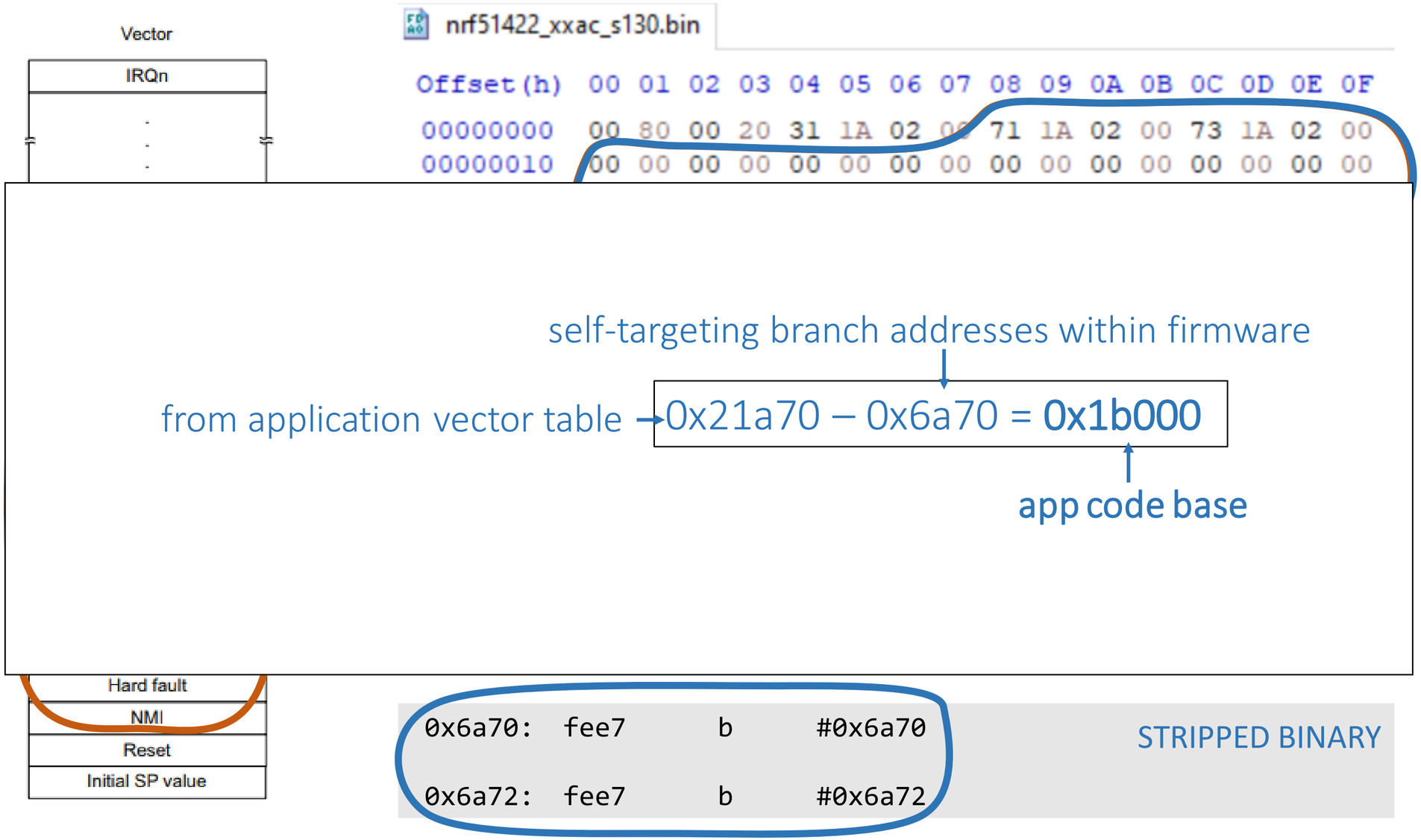
UNSTRIPPED BINARY

```
00021a70 <NMI_Handler>:
   21a70: e7fe      b.n21a70 <NMI_Handler>

00021a72 <HardFault_Handler>:
   21a72: e7fe      b.n21a72 <HardFault_Handler>
```

STRIPPED BINARY

```
0x6a70:  fee7      b      #0x6a70

0x6a72:  fee7      b      #0x6a72
```

1. Identify app code base



2. Identify inline data

LDR and variants

```
0x1b8a2: 084b          ldr      r3, [pc, #0x20] ; 0x1b8c4
 ...
0x1b8c4: 6c21          movs     r1, #0x6c
0x1b8c6: 0020          movs     r0, #0
```

TBB and TBH

```
0x2a6e0: 2b04          cmp      r3, #4
0x2a6e2: d82d          bhi.n    2a740 <characteristic_add+0x16c>
0x2a6e4: e8df f003     tbb      [pc, r3]
0x2a6e8: 899d9995      ldmibhi  sp, {r0, r2, r4, r7, r8, fp, ip, pc}
0x2a6ec: 23220029      ; <UNDEFINED> instruction: 0x23220029
```

__ARM_common_switch8 calls

```
0x1ca5e: f7fefd49      bl       1b4f4 <__ARM_common_switch8>
0x1ca62: 15150d18      ldrne    r0, [r5, #-3352] ; 0xfffff2e8
0x1ca66: 67554737      smmlarvs r5, r7, r7, r4
0x1ca6a: 887f887c      ldmdahi  pc!, {r2, r3, r4, r5, r6, fp, pc}^  ; <UNPREDICTABLE>
0x1ca6e: 9e9ede91      mrcls    14, 4, sp, cr14, cr1, {4}
0x1ca72: c89e9e9e      ldmgt    lr, {r1, r2, r3, r4, r7, r9, sl, fp, ip, pc}
0x1ca76: ded8d1ce      atnle    <illegal precision>m   f5, #0.5
0x1ca7a: 210f14de      ldrdcs   r1, [pc, -lr]
0x1ca7e: 9a02          ldr      r2, [sp, #8]
```

data

3. Estimate function blocks

Function start addresses = {0x21A70, 0x21A72, 0x21A74, 0x21A76, 0x21A78, 0x21A7A }

Known function start addresses

nrf51422_xxac_s130.bin

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00000000	00	80	00	20	31	1A	02	00	71	1A	02	00	73	1A	02	00
00000010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000020	00	00	00	00	00	00	00	00	00	00	00	00	75	1A	02	00
00000030	00	00	00	00	00	00	00	00	77	1A	02	00	79	1A	02	00
00000040	7B	1A	02	00	7B	1A	02	00	61	C5	01	00	7B	1A	02	00
00000050	7B	1A	02	00	00	00	00	00	45	C0	01	00	7B	1A	02	00
00000060	7B	1A	02	00	7B	1A	02	00	7B	1A	02	00	7B	1A	02	00
00000070	7B	1A	02	00	7B	1A	02	00	7B	1A	02	00	7B	1A	02	00
00000080	7B	1A	02	00	B9	B4	01	00	7B	1A	02	00	7B	1A	02	00
00000090	95	B5	01	00	7B	1A	02	00	A1	19	02	00	7B	1A	02	00

3. Estimate function blocks

Function start addresses = {0x1B198, 0x1B1C4, 0x1E87C, 0x21A70, 0x21A72, 0x21A74, 0x21A76, 0x21A78, 0x21A7A }

High-certainty function starts [bl-targets]

```
1b160:  480c       ldr   r0, [pc, #48] ;
1b162:  e000       b.n   1b166
1b164:  bf00       nop
1b166:  f000 f82d  bl 1b1c4
1b16a:  0020       movs  r0, r4
1b16c:  0029       movs  r1, r5
1b16e:  f003 fb85  bl 1e87c
1b172:  f000 f811  bl 1b198
1b176:  46c0       nop        ; (mov r8, r8)
```

3. Estimate function blocks

Function start addresses = {0x1B198, 0x1B1BC, 0x1B1C4, 0x1E87C, 0x21A70, 0x21A72, 0x21A74, 0x21A76, 0x21A78, 0x21A7A }

Function estimation algorithm

```
0x1b198: push {r4, lr}
0x1b19a: cmp r2, #0x20
0x1b19c: blt #0x1b1a8
0x1b19e: mov r0, r1
0x1b1a0: subs r2, #0x20
0x1b1a2: lsrs r0, r2
0x1b1a4: movs r1, #0
0x1b1a6: pop {r4, pc}
0x1b1a8: mov r3, r1
0x1b1aa: lsrs r3, r2
0x1b1ac: lsrs r0, r2
0x1b1ae: movs r4, #0x20
0x1b1b0: subs r2, r4, r2
0x1b1b2: lsls r1, r2
0x1b1b4: orrs r0, r1
0x1b1b6: mov r1, r3
0x1b1b8: pop {r4, pc}
0x1b1ba: nop
0x1b1bc: ldrb r2, [r0]
```

4. Pinpoint instructions of interest

```
uint8_t passkey[] = "123456";
ble_opt_t ble_opt;
ble_opt.gap_opt.passkey.p_passkey = &passkey[0];
err_code = sd_ble_opt_set(BLE_GAP_OPT_PASSKEY, &ble_opt);
```

4. Pinpoint instructions of interest

- ARM supervisor calls
  - Easy to identify within disassembly (instruction with opcode `svc`)
  - `svc` numbers of interest can be found within vendor SDKs
- Function calls
  - argXtract uses *function pattern files* to identify functions of interest, i.e., files representing expected outputs at specific addresses for given inputs
  - Function pattern file is passed to the set of identified functions. Each function is executed with the inputs in the function pattern file, and the outputs are compared with the expected values
  - Assumes that the function of interest produces unique, distinguishable outputs or artefacts

5. Generate trace chains and trace through assembly

- Register and memory contents are stored and processed during the trace
- Upon reaching the function/svc of interest, register+memory contents are returned for analysis

6. Process register/memory objects

Trace object

```
{'sd_ble_opt_set':{
'memory':{...,
   20007f60:'31',
   20007f61:'32',
   20007f62:'33',
   20007f63:'34',
   20007f64:'35',
   20007f65:'36',
   20007f66:'00',
   20007f68:'60',
   20007f69:'7f',
   20007f6a:'00',
   20007f6b:'20',
   20007f6c:'01',
   ...},
'registers':{...,
   'pc':'0001ea96',
   'sp':'20007f48',
   'r0':'00000022',
   'r1':'20007f68',
   ...}
  }
}
```

Argument definition object

```
{"args": {
  "0": {...},
  "1": {
   "in_out": "in",
   "ptr_val": "pointer",
   "data": {
      "p_opt": {
         "ptr_val": "pointer",
         "length_bits": 48,
         "type": "hex“
      }
   }
  }
}}
```

Output

```
 "output": {
   "sd_ble_opt_set": [{
      "opt_id": 34,
      "p_opt": "313233343536"
   }]
}
```

Hex for “123456”

Evaluation: Test Set

- No existing ground truth for Cortex-M
- Manually generated test set
- 28 binaries
- Chipset vendors: NXP, STMicroelectronics, Nordic Semiconductor and Texas Instruments
- Technologies: Zigbee, ANT, BLE, Thread and 802.15.4
- Compilers: GCC, IAR, Keil and Clang (depending on options made available by chipset vendors)

Evaluation: Function Identification

Bin File		argXtract		radare2		ghidra		Bin File		argXtract		radare2		ghidra	
ID	#Fns	TPR	EFPR	TPR	EFPR	TPR	EFPR	ID	#Fns	TPR	EFPR	TPR	EFPR	TPR	EFPR
0a1	324	100	0.29	95.68	2.19	87.96	0	0d2	841	0	100	69.32	7.4	69.68	2.85
1d7	951	93.27	0.97	74.24	3	73.08	0.82	3b1	204	99.02	0	88.24	0	82.35	0
443	598	100	0	83.78	3.05	83.95	1.95	4d7	1563	95.71	1.17	78.57	3.28	77.8	0.15
589	1486	97.51	0.68	83.24	1.59	84.79	0	5d3	398	99.5	0.73	94.97	0	93.47	0
646	166	98.8	0	80.72	0.73	77.71	0.76	67e	2138	99.16	0.05	82.69	0.54	83.4	0.22
681	1961	97.86	0.56	94.19	0.7	87.51	0.12	6ac	265	98.11	0.37	73.96	0.5	72.08	0
70b	115	95.65	0	67.83	0	73.04	0	7e8	1529	97.58	0.66	81.62	1.57	84.96	0
928	520	95.38	0.93	90.19	0	71.15	0	938	2764	99.57	0.74	85.53	0.59	83.9	0.09
989	762	95.8	9.7*	69.27	8.62	63.1	9.39	ade	1951	99.33	0.89	89.54	0.46	87.69	0.12
bad	839	92.25	0.79	69.85	5.43	68.53	0	be7	2035	99.71	0.39	5.11	87.43	4.72	88
cb5	92	94.57	1.11	61.96	0	67.39	0	cc8	1582	94.82	0.71	82.68	1.91	83	0
dd9	801	96.63	6.9*	95.93	6.36	88.15	6.66	e2a	495	95.15	0.39	89.7	0	69.29	0
e2d	698	96.42	0.35	94.99	0	86.25	0	f2b	1926	99.79	0.65	81.15	1.01	79.85	0.06
f37	1585	95.21	1.16	78.23	3.18	78.36	0	fe9	1007	99.4	0.1	61.27	0.8	56.21	0.7

Case Studies

Case Study	Chipset Vendor	Technology	Config Mechanism	Tested Security Metrics	Tested Calls of Interest
Nordic BLE (243)	Nordic Semiconductor	Bluetooth Low Energy	Supervisor calls	Security of BLE Data	`sd_ble_gatts_characteristic_add, sd_ble_gatts_service_add`
				Use of Fixed Passkeys	`sd_ble_opt_set`
				User Tracking due to Fixed Addresses	`sd_ble_gap_address_set, sd_ble_gap_addr_set, sd_ble_gap_privacy_set`
				Manufacturer/Device Names and Privacy	`sd_ble_gap_device_name_set`
BlueNRG (2)	STMicroelectronics	Bluetooth Low Energy	Function calls	BLE Address Privacy	`aci_hal_write_config_data, aci_gap_init`
				BLE Pairing Security	`aci_gap_set_io_capability, aci_gap_set_authentication_requirement, aci_gap_set_authorization_requirement`
ANT (9)	Nordic Semiconductor	ANT	Supervisor calls	ANT Channel Security	`sd_ant_enable`

Findings

- Over 50% of binaries had little to no protection for vendor-defined data (including ECG monitors, HID devices, smart watches, safety aids…)
- Inconsistent application of permissions to restrict access to data
- Use of fixed pairing passkeys, i.e., weakened pairing mechanism
- 95% of devices – including all wearables – use fixed device addresses, enabling device (and potentially also user) tracking

Conclusions

- Framework to extract security configurations from stripped Cortex-M binaries
- Several vulnerabilities identified in real-world IoT devices
- Still work to be done: improving function pattern matching
- Ground truth!

Thank You