WIP: Verified parser generator for microcontroller applications

Sameed Ali
sameed.ali.gr@dartmouth.edu
Sean Smith
sws@cs.dartmouth.edu

DARTMOUTH

FDA NEWS RELEASE

FDA Informs Patients, Providers and Manufacturers About Potential Cybersecurity Vulnerabilities in Certain Medical Devices with Bluetooth Low Energy

For Immediate Release: March 03, 2020

By Joe Carlson Star Tribune | MARCH 6, 2020 — 7:36PM

Recent BLE vulnerability disclosures

CVE List ▾ CNAs ▾ WGs ▾ Board ▾ About ▾ News & Blog ▾

erabilities and Exposures

Request CVE IDs Upd

TOTAL CVE

VE > SEARCH RESULTS

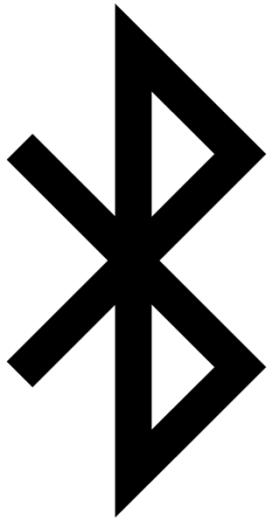
h Results

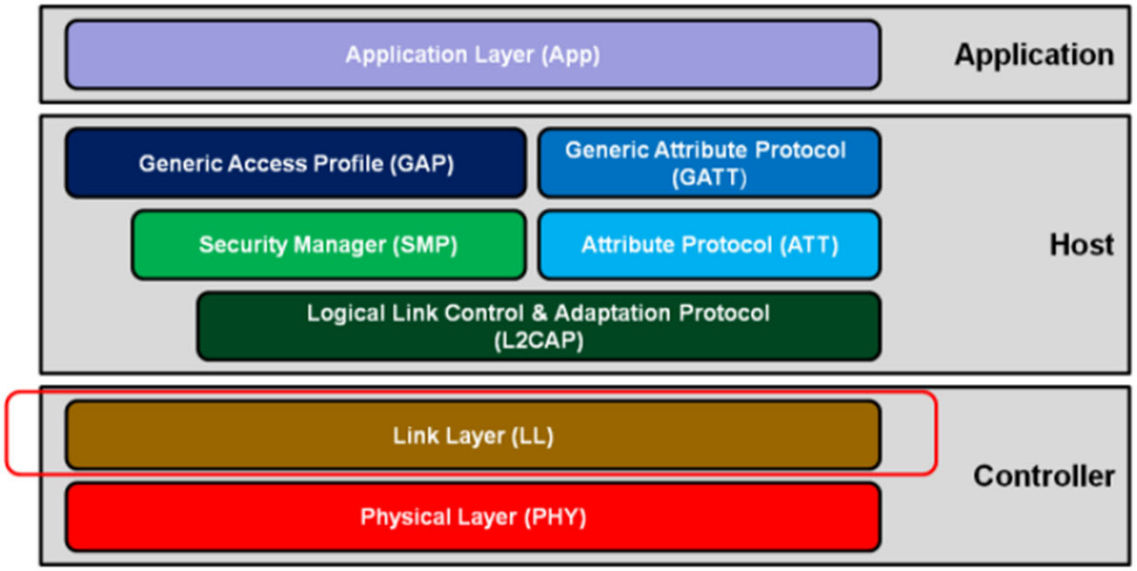
39 CVE entries that match your search.

me	Description
D-15582	An issue was discovered on Samsung mobile devices with P(9.0) and Q(10.0) (Exynos 7885 chipsets) software. The Bluetooth Low Energy (BLE) component has a l wi[illegible]
D-15509	Nc[illegible] an engage i co[illegible] leManagerH
D-13595	Th[illegible] leted BLE pa tri[illegible] rtion (which ta[illegible]
D-13594	Th[illegible] el map field re[illegible]
D-12860	CC[illegible] DSafe uses allows for re-identification of a device, and potentially identification of the owner's name.
D-11957	The Bluetooth Low Energy implementation in Cypress PSoC Creator BLE 4.2 component versions before 3.64 generates a random number (Pairing Random) with sig entropy than the specified 128 bits during BLE pairing. This is the case for both authenticated and unauthenticated pairing with both LE Secure Connections as well Pairing. A predictable or brute-forceable random number allows an attacker (in radio range) to perform a MITM attack during BLE pairing.
D-10685	A flaw was found in Ansible Engine affecting Ansible Engine versions 2.7.x before 2.7.17 and 2.8.x before 2.8.11 and 2.9.x before 2.9.7 as well as Ansible Tower be versions 3.4.5 and 3.5.5 and 3.6.3 when using modules which decrypts vault files such as assemble, script, unarchive, win_copy, aws_s3 or copy modules. The tem is created in /tmp leaves the s ts unencrypted. On Operating Systems which /tmp is not a tmpfs but part of the root partition, the directory is only cleared on boot emains when the host is switched off. The system will be vulnerable when the system is not running. So decrypted data must be cleared as soon as possible and th normally is encrypted ble.
D-0129	In SetData of btm_ble_multi_adv.cc, there is a possible out-of-bound write due to an incorrect bounds check. This could lead to local escalation of privilege with no execution privileges needed. User interaction is not needed for exploitation.Product: AndroidVersions: Android-10Android ID: A-123292010

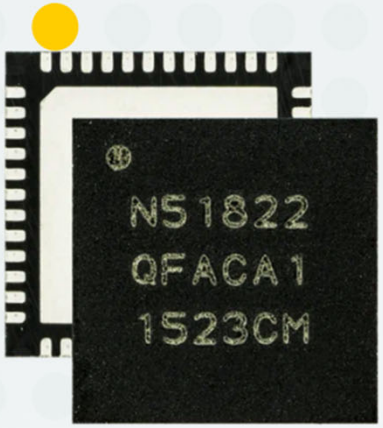
SweynTooth Bluetooth LE Security Flaw Threatens Thousands Of Devices From Samsung, FitBit And More

Bluetooth Low Energy Protocol





BLE Controller SoC's



QFN48 6x6mm

QFN48 WLCSP

nRF51822

System on Chip

Bluetooth Low Energy and 2.4 GHz SoC

The nRF51822 is a general purpose, ultra-low power SoC ideally suited for Bluetooth® Low Energy and 2.4 GHz proprietary wireless applications. It is built around the 32-bit ARM® Cortex™-M0 CPU with 256/128 KB flash and 32/16 KB RAM. The flexible 2.4 GHz radio supports Bluetooth Low Energy and 2.4 GHz proprietary protocols, such as Gazell.

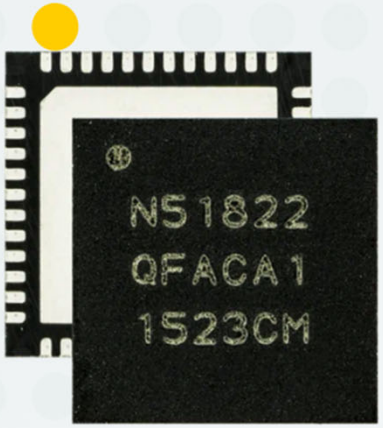
It incorporates a rich selection of analog and digital peripherals that can interact directly without CPU intervention

Contact us about this

Key Features

16 MHz **Cortex-M0**
256/128 KB **Flash**,
32/16 KB **RAM**
2.4 GHz Transceiver
2 Mbps, 1 Mbps, 250 kbps
Bluetooth Low Energy
+4 dBm TX Power
128-bit AES CCM
UART, SPI, TWI

BLE Controller SoC's



QFN48 6x6mm

QFN48 WLCSP

nRF51822

System on Chip

Bluetooth Low Energy and 2.4 GHz SoC

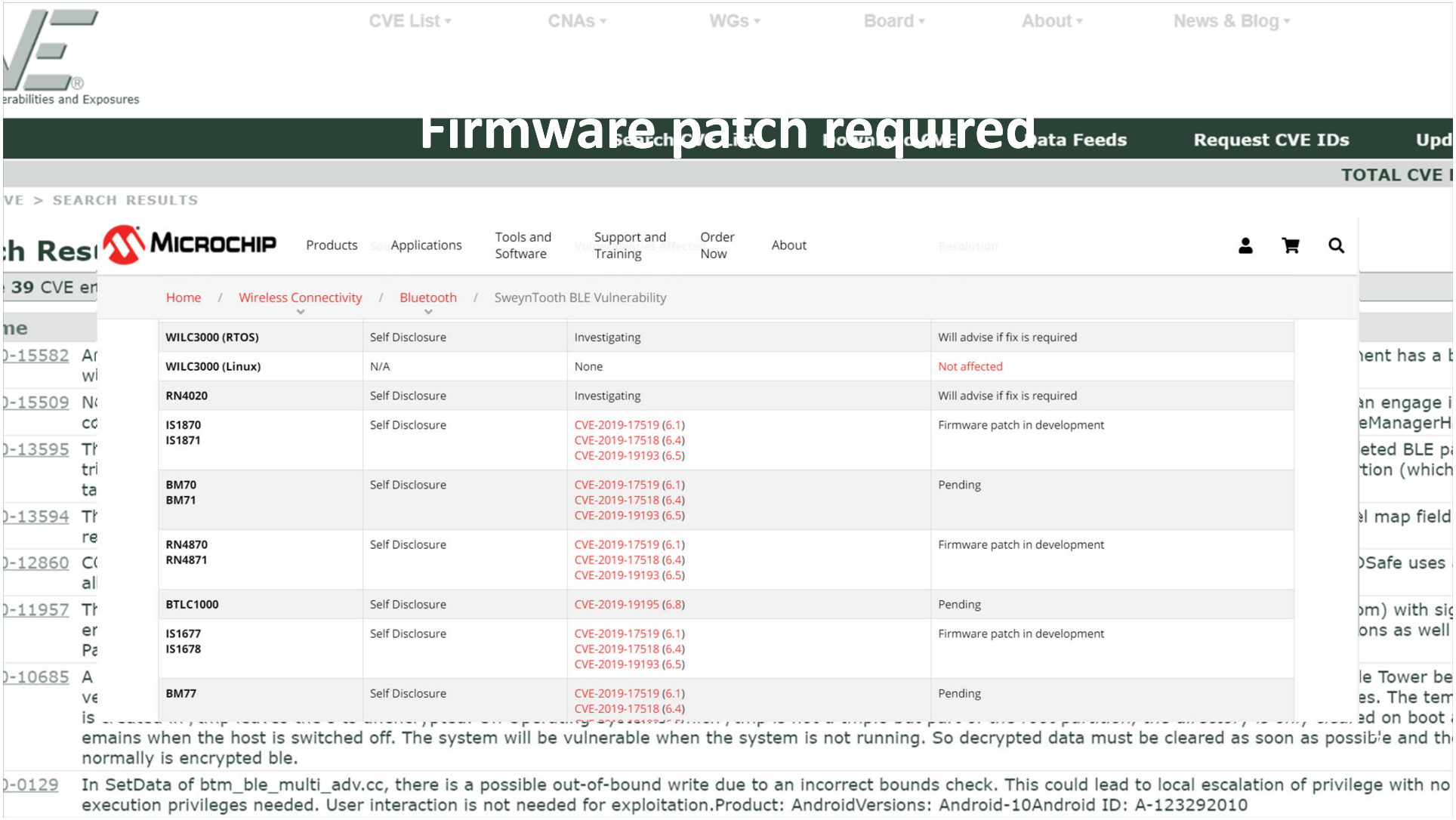
The nRF51822 is a general purpose, ultra-low power SoC ideally suited for Bluetooth® Low Energy and 2.4 GHz proprietary wireless applications. It is built around the 32-bit ARM® Cortex™-M0 CPU with 256/128 KB flash and 32/16 KB RAM. The flexible 2.4 GHz radio supports Bluetooth Low Energy and 2.4 GHz proprietary protocols, such as Gazell.

It incorporates a rich selection of analog and digital peripherals that can interact directly without CPU intervention

Contact us about this

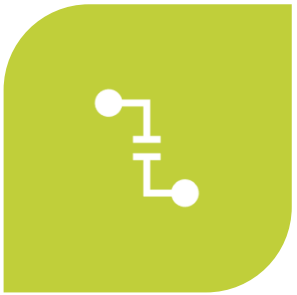
Key Features

16 MHz **Cortex-M0**
256/128 KB **Flash**,
32/16 KB **RAM**
2.4 GHz Transceiver
2 Mbps, 1 Mbps, 250 kbps
Bluetooth Low Energy
+4 dBm TX Power
128-bit AES CCM
UART, SPI, TWI


CVE List
CNAs
WGs
Board
About
News & Blog
erabilities and Exposures
Firmware patch required
Data Feeds
Request CVE IDs
TOTAL CVE
VE > SEARCH RESULTS
MICROCHIP
Products
Applications
Tools and Software
Support and Training
Order Now
About
Home / Wireless Connectivity / Bluetooth / SweynTooth BLE Vulnerability
WILC3000 (RTOS) | Self Disclosure | Investigating | Will advise if fix is required
WILC3000 (Linux) | N/A | None | Not affected
RN4020 | Self Disclosure | Investigating | Will advise if fix is required
IS1870 IS1871 | Self Disclosure | CVE-2019-17519 (6.1) CVE-2019-17518 (6.4) CVE-2019-19193 (6.5) | Firmware patch in development
BM70 BM71 | Self Disclosure | CVE-2019-17519 (6.1) CVE-2019-17518 (6.4) CVE-2019-19193 (6.5) | Pending
RN4870 RN4871 | Self Disclosure | CVE-2019-17519 (6.1) CVE-2019-17518 (6.4) CVE-2019-19193 (6.5) | Firmware patch in development
BTLC1000 | Self Disclosure | CVE-2019-19195 (6.8) | Pending
IS1677 IS1678 | Self Disclosure | CVE-2019-17519 (6.1) CVE-2019-17518 (6.4) CVE-2019-19193 (6.5) | Firmware patch in development
BM77 | Self Disclosure | CVE-2019-17519 (6.1) CVE-2019-17518 (6.4) | Pending
emains when the host is switched off. The system will be vulnerable when the system is not running. So decrypted data must be cleared as soon as possible and th
normally is encrypted ble.
In SetData of btm_ble_multi_adv.cc, there is a possible out-of-bound write due to an incorrect bounds check. This could lead to local escalation of privilege with no
execution privileges needed. User interaction is not needed for exploitation.Product: AndroidVersions: Android-10Android ID: A-123292010

Objectives

DEVELOP A FRAMEWORK FOR QUICKLY GENERATING HARDENED PARSERS FOR VARIOUS BINARY PROTOCOLS

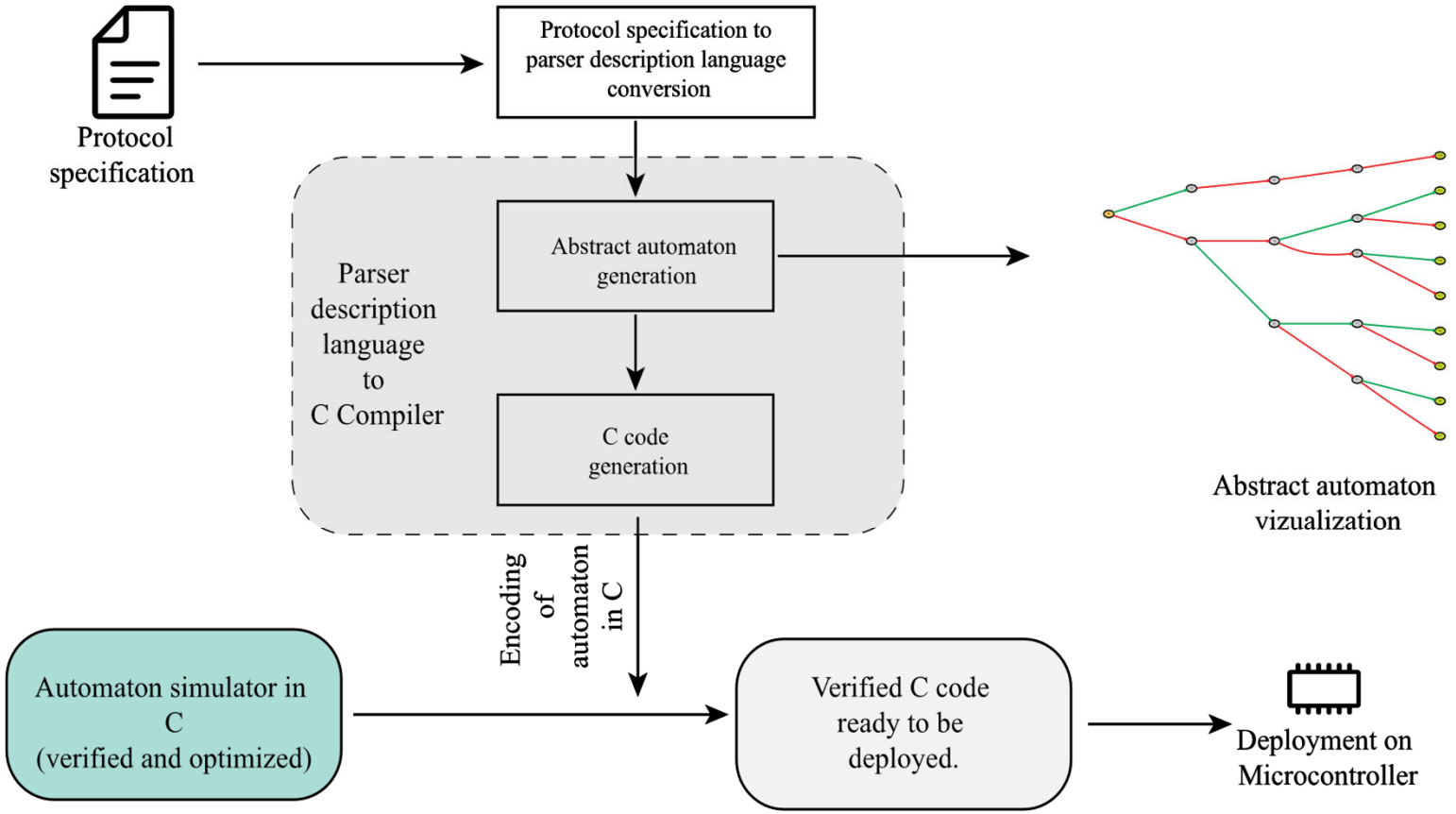


DEVELOP HARDENED PARSERS WHICH CAN OPERATE EFFECTIVELY ON A RESOURCE CONSTRAINED MICROCONTROLLERS

Ensure the hardened parsers:

- Are backed by a formal language model (finite state machine)
- Are free-from memory corruption vulnerabilities.
- Terminate on all supplied inputs.
- Can operate with the limited resources of a microcontroller.

Architecture Diagram



Define a Parser Description Language

Has constructs commonly found in binary protocols like Tag-length-value, Repeat etc.

Is composable, and readable

Allows quick prototyping of different binary protocols

Example of Parser Description Language

- The PDL has constructs corresponding to commonly found constructs in the binary protocols like Tag-Length-Value etc.
- Example on right shows a BLE LL header description.

```
;; clstp the header packet for  ADV packets
(def PDU
  (gt/gen-tag 4
              {:PDU=ADV_IND         "0000"
               :PDU=ADV_DIRECT_IND  "0001"
               :PDU=ADV_NONCONN_IND "0010"
               :PDU=SCAN_REQ        "0011"
               :PDU=SCAN_RSP        "0100"
               :PDU=CONNECT_IND     "0101"
               :PDU=ADV_SCAN_IND    "0110"
               :PDU=ADV_EXT_IND     "0111"
               :PDU=AUX_CONNECT_RSP "1000"}
              "PDU"))

(def RFU   (gt/gen-tag 1 {:RFU=RFU_ON    "1"   :RFU=RFU_OFF      "0"}
(def ChSel (gt/gen-tag 1 {:CHSEL=CHSEL_ON  "1" :CHSEL=CHSEL_OFF  "0"}
(def TxAdd (gt/gen-tag 1 {:TXADD=TX_ADD_ON "1" :TXADD=TX_ADD_OFF "0"}
(def RxAdd (gt/gen-tag 1 {:RXADD=RX_ADD_ON "1" :RXADD=RX_ADD_OFF "0"}

(def len-field
  (gc/gen-len 8 :LSB "header_len"))

;; header is a sequence of these automatons
(def adv-packet-header
  (reduce os/seq-graphs
          [PDU RFU ChSel TxAdd RxAdd len-field]))
```

Verification of C code

Verification of C code done via Frama-C (Static analyzer for C code)

Code annotations of pre/post conditions and invariants ensure **termination** and **no memory corruption.**

Verified code is a FSM simulator which takes an encoding of the abstract FSM, and an input to the FSM as input, and simulates it.

Preliminary results

Deployed hardened parsers for BLE LL on Ubertooth One device

Attacked the device with malformed BLE packets

The parsers successfully rejected the malformed packets

Questions?

Thank you!