SHiFT: Semi-hosted Fuzz Testing for Embedded Applications

Alejandro Mera, Changming Liu, Ruimin Sun, Engin Kirda, Long Lu

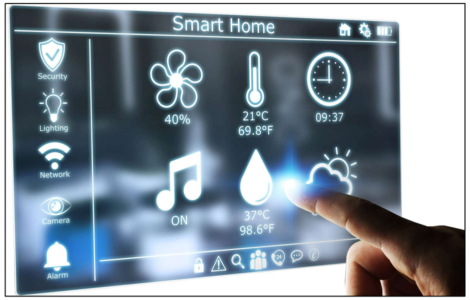
Northeastern University

FLORIDA INTERNATIONAL UNIVERSITY

Northeastern University
Khoury College of Computer Sciences

Embedded devices in the IoT era

Embedded devices are everywhere and adopted in critical areas



"IoT adoption is critical to ongoing business success[1]"

[1] The State of IoT/OT Cybersecurity in the Enterprise, Ponemon Institute, 2021.

Vulnerabilities and Attacks on Embedded Devices

The New York Times

A New Era of Internet Attacks Powered by Everyday Devices

DARKReading | The Edge | DR Tech | Sections | Events

Medical and IoT Devices From More Than 100 Vendors Vulnerable to Attack

PC | #ThePCMagCheap100 | #Windows11 | Reviews | Best Products | How-To | News | Newsletters

Home > News > Security

CISA Warns That BrakTooth Vulnerabilities Can Now Be Exploited

A proof of concept exploit for the BrakTooth flaw in countless Bluetooth devices has been shared.

BIZ & IT —

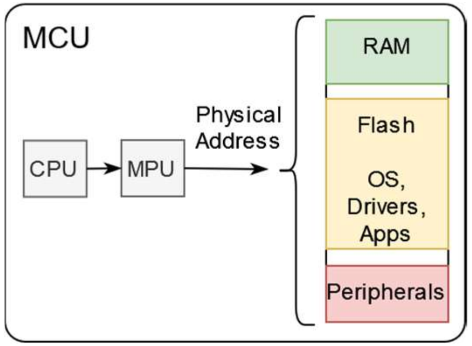
Broadcom chip bug opened 1 billion phones to a Wi-Fi-hopping worm attack

Wi-Fi chips used in iPhones and Android may revive worm attacks of old.

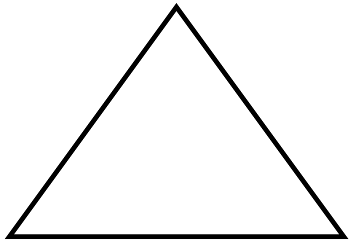
Challenges testing embedded devices

Diversity of
Software and Hardware

Microcontrollers

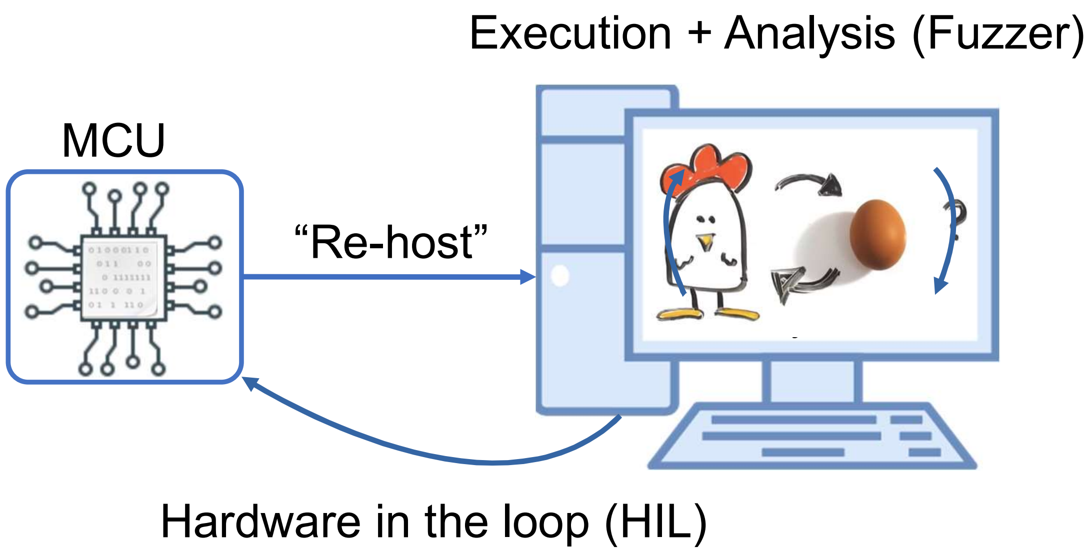


Minimalistic
design



Computing and
operational constraints

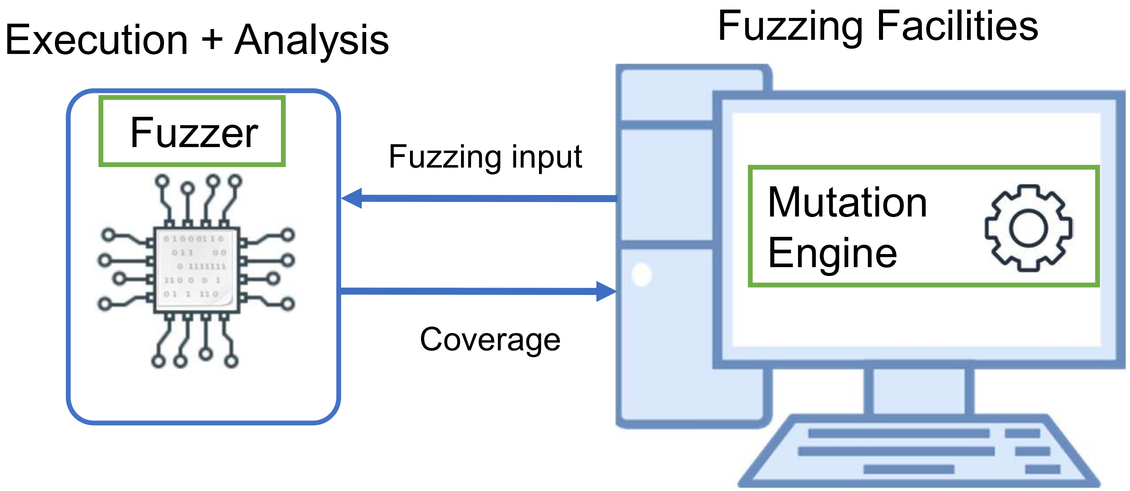
The state of the art: Re-hosting



Open Challenges:

- Reduced Fidelity
- Reduced observability
- Limited compatibility

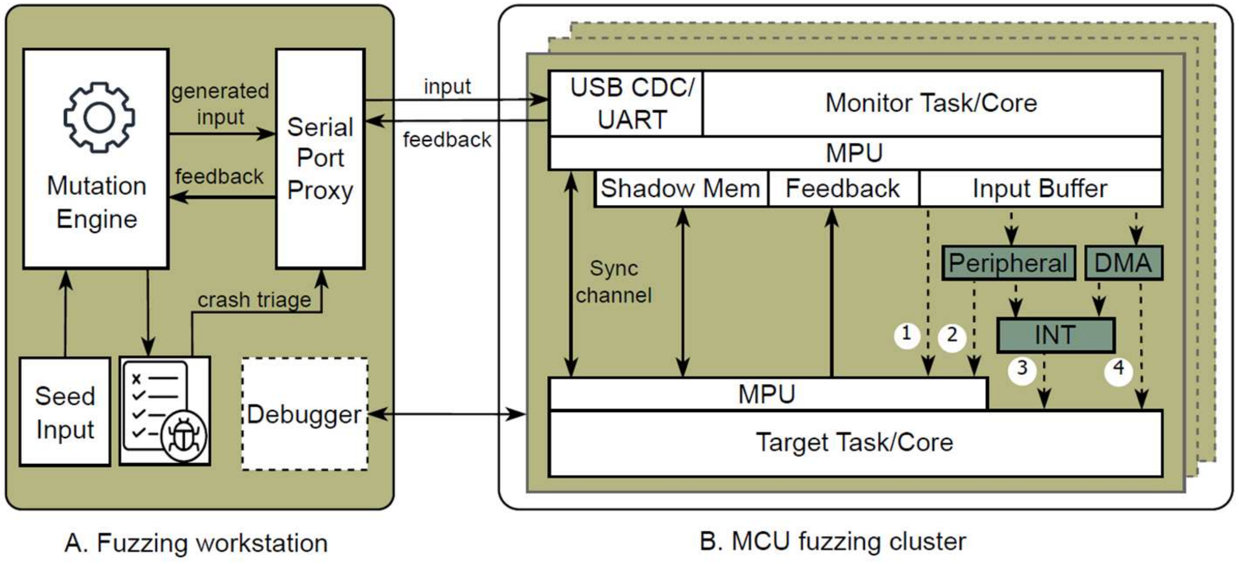
SHiFT: Semi-Hosted Fuzz Testing



"Semihosting enables firmware, running natively in an MCU, to use facilities available in a workstation"

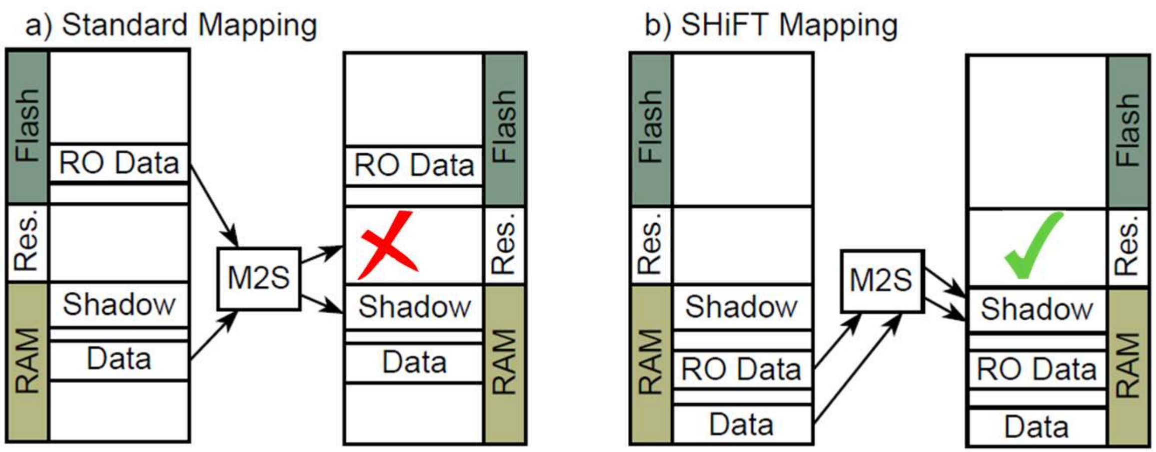
Northeastern University
Khoury College of Computer Sciences

Design: SHiFT proposed architecture



- **Design goals:**
  - Meant for in-house testing
  - Supports desktop-level instrumentation
  - Compatible with standard development platforms

Design: supporting ASAN instrumentation without MMU



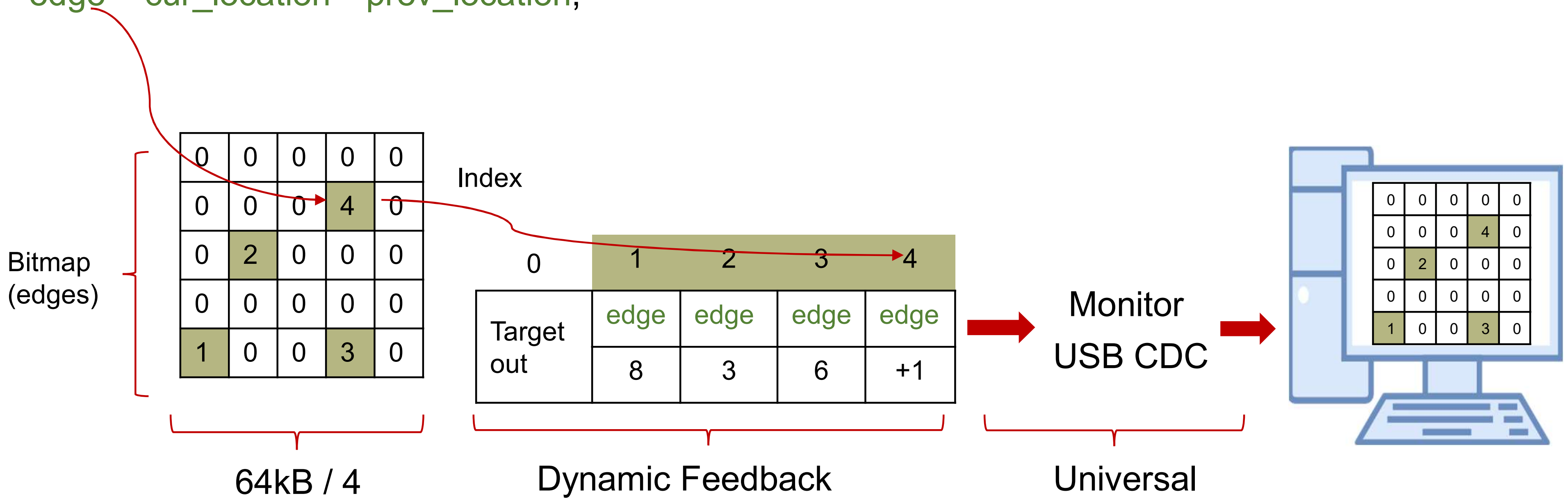
ASAN Mem-to-Shadow (M2S) : (Addr>>3) + Offset
Incompatible with MCU (Muench et al. 2018)

Faults
- Cortex-M MPU and traps
- Exception model

Instrumentation
- Memory map relocation
- Tailored offsets

Design: Coverage, feedback and communication protocol

SANCOV:
cur_location = PC;
edge = cur_location ^ prev_location;



Implementation:

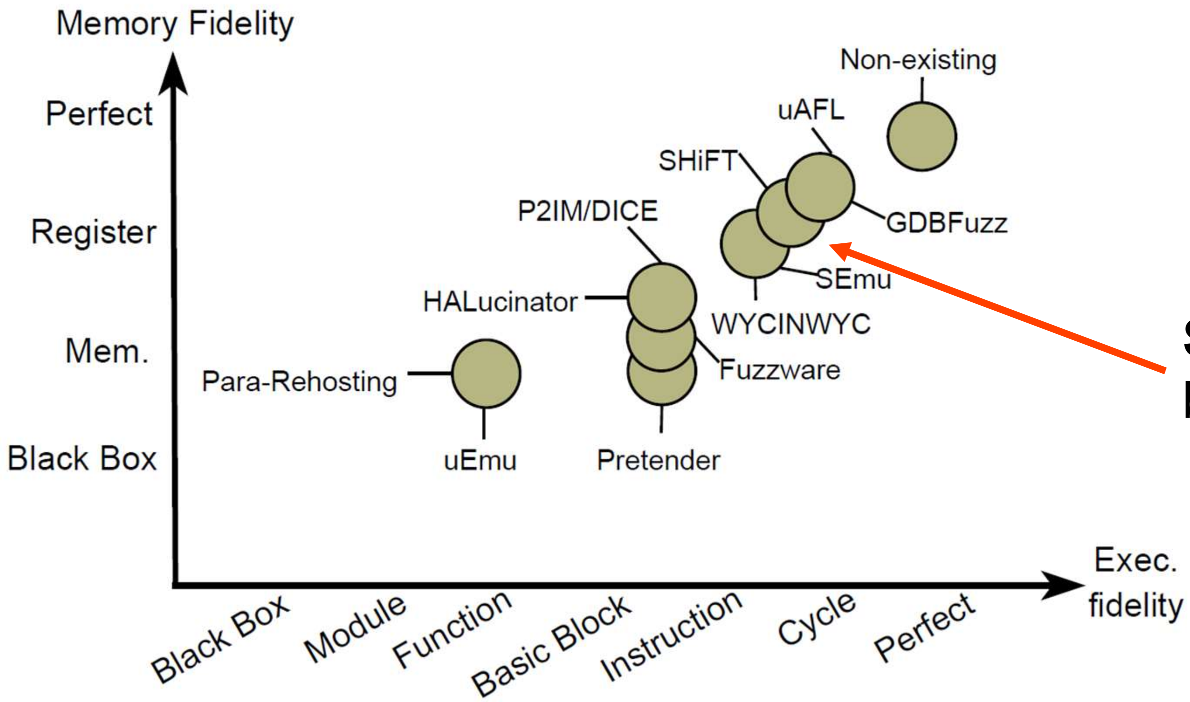
- ARMv7-M, ARMv8-M and Xtensa
- Universal serial Proxy (AFL/AFL++ and others)
- Firmware (MCU)
  - FreeRTOS 10.4 Kernel extensions
  - Instrumentation runtime GNU ARM 10.3
  - Tailored GCC compiler (ASAN offset)

Evaluation: Architecture compatibility

Architecture	MPU	GCC	SANCOV	ASAN	UBSAN	Port MCU
ARMv7-M	✓	9.3.1	✓	✓	✓	SMT32H745/H743 SAMD51, K66F
ARMv8-M	✓	9.3.1	✓	✓	✓	STM32L552
Xtensa	✓	8.4.0	✓	✓	✓	ESP32 WROM
MIPS M4K	MMU	8.3.1	✓		✓	PIC32MX795
MIPS MK64F	MMU	8.3.1	✓		✓	PIC32MZ2048
RISC-V	optional	9.2.0	✓		✓	GD32VF103CBT6
Renesas RX	✓	8.3	✓		✓	RSF562N8BDFP
Renesas RL	✓	11.1*	✓		✓	–
AVR		7.3.0	✓			Atmega2560
MSP430	optional	9.3.1	✓			–
ARC	optional	11.2.0	✓			–
Coldfire		9.3.0	✓			–
Power PC 400		9.3.0	✓	✓	✓	–

Fully compatible with **12 embedded** architectures

Evaluation: Fidelity analysis



Superior to all emulation-based solutions

Based on the 2-dimensional analysis proposed by Wright et al., 2021

Evaluation: synthetic raw performance

Fuzzing Mode	Native AFL	SHiFT S-C	SHiFT D-C
Standard	4.9	4.8	0.41
Persistent	23.5	5.9	5.1
Standard With ASAN	*1.9*	*4.6*	*0.32*
Persistent With ASAN	22.7	5.7	4.1

Performance in [kRun/s] of a single instance of SHiFT for single (S-C) and dual-core (D-C) configurations compared with native AFL (Ubuntu 22.04, AMD Ryzen 3700x, 32 GB).

2.4x faster than a workstation

Evaluation: case studies and comparison with the SoTA

5 new vulnerabilities

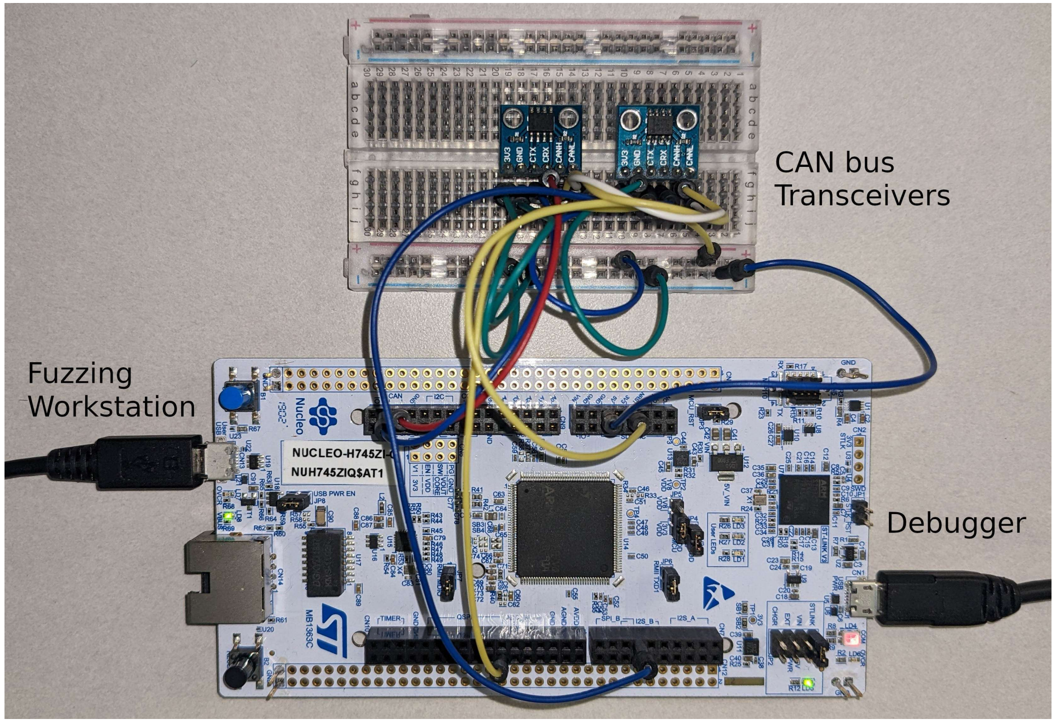
No false positives

~100x faster

Ref	#	Firmware	Method	Board	SHiFT [r/s]	SHiFT TP	SHiFT FP	P2IM/DICE [r/s]	P2IM/DICE SU	P2IM/DICE TP	P2IM/DICE FP	Fuzzware [r/s]	Fuzzware SU	Fuzzware TP	Fuzzware FP	GDBFuzz [r/s]	GDBFuzz SU	GDBFuzz TP	GDBFuzz FP
P2IM	1	PLC	Function call	H743	3100	4	0	32.7	×95	4	4	30.9	×100	4	2	70	×44	4	0
DICE	2	Modbus	Full-stack DMA	H743	1800	3	0	41.6	×43	3	2	NB	-	-	-	327	×6	0	0
	3	Midi	Full-stack DMA	H743	1200	2	0	59.9	×20	2	0	208	×6	0	0	37	×32	0	0
SHiFT	4	Synthetic	Function call	H743	4800	11	0	94.5	×51	3	1	85.9	×55	0	10	32.1	×150	2	0
	5	GPS Receiver	Function call	ESP32	380	0	0	NB	-	-	-	NB	-	-	-	170	×2	0	0
	6	AT parser	Function call	SAMD51	276	0	0	44.1	×6	0	1	53.5	×5	0	1	55	×5	0	1
	7	Command line	Function call	K66F	233	0	0	63.5	×4	0	1	321.9	×1	0	1	245	×1	0	1
	8	Shelly Dimmer	Real-time DMA	H743	1148	3	0	NB	-	-	-	321.3	×4	0	1	24.5	×25	0	4
	9	Bootloader	Baremetal	H745	170	1	0	NB	-	-	-	89	×2	0	0	NB	-	-	-
	10	FreeRTOS K.	Function call	L552	3750	1	0	NB	-	-	-	NB	-	-	-	43	×86	0	0

24-hour fuzzing campaigns of SHiFT on real firmware and a Synthetic benchmark compared to the SoTA. SU: SHiFT SpeedUp (average), TP: TruePositives (median), FP: False Positives (median), NB: No Bootstrap. New TPs observed on firmware # 8, 9, 10.

Evaluation: testing capabilities analysis (CAN bus)



- Supports complex peripherals not supported by emulators.
- Great flexibility to leverage heterogenous architectures (M7 & M4)
- Holistic considering operative and timing constraints

Evaluation: unique testing capabilities on real scenarios

Fidelity (real time operations)

Firmware	Vulnerability	CWE	Instances
Shelly Dimmer	Divide by zero	369	3
Bootloader	Buffer overflow	120	1
FreeRTOS K.	Improper handling of insufficient privileges	274	1

Observability (Instrumentation)

Compatibility (Complex peripherals, e.g. CAN)

Conclusions

- **Testing** embedded devices require holistic methods that consider SW and HW diversity, minimalistic designs, and operational constraints.
- SHiFT is a novel semihosted framework to enable fuzz testing on development platforms with high fidelity.
- SHiFT helped to identify unknown vulnerabilities in realistic scenarios not supported by emulation-based solutions.

Northeastern University
Khoury College of Computer Sciences

SHiFT: Semi-hosted Fuzz Testing for Embedded Applications

Alejandro Mera

Source code: https://github.com/RiS3-Lab/SHiFT

Thanks!