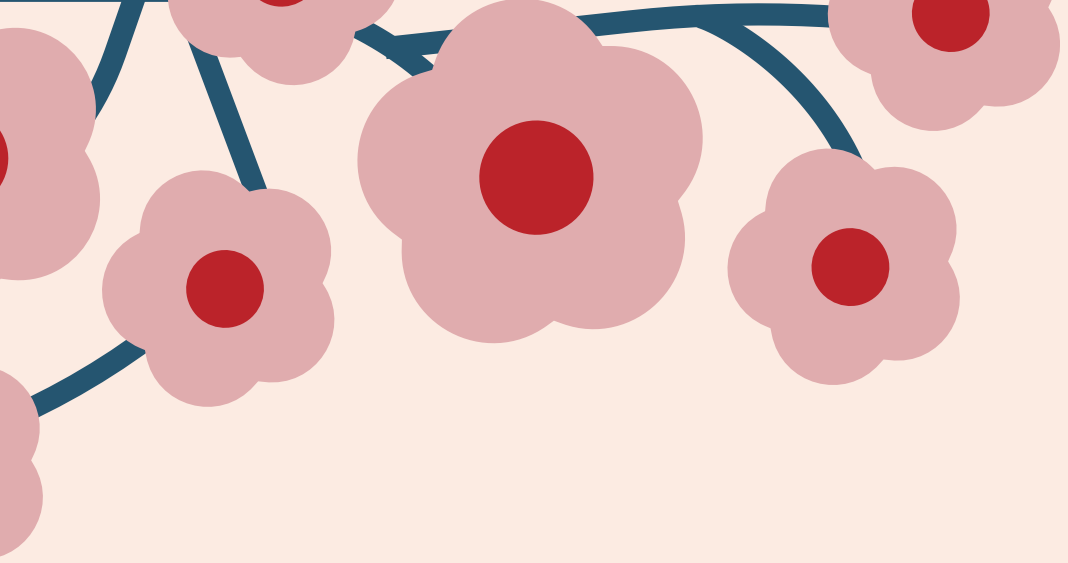
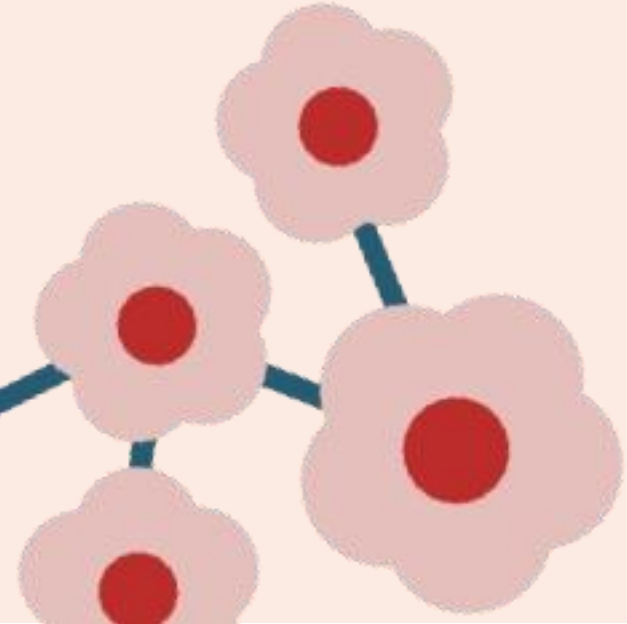
2 April 2025



CHERI

CHERI MicroPython

Challenges and Opportunities



Jeremy Singer
University of Glasgow

What is MicroPython?

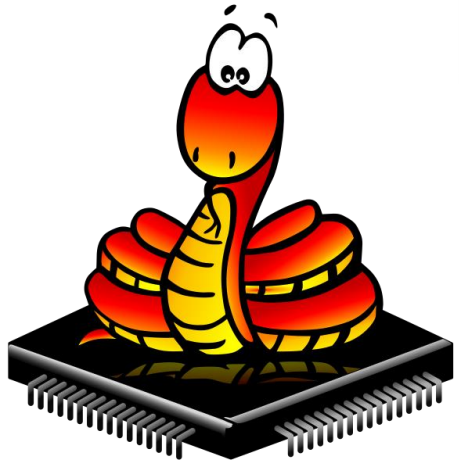
A cut–down user–friendly Python **interpreter**

Designed for resource constrained **micro–controllers**

Supports most of Python3

Extra libraries for **hardware** control

Ports available for Linux / **RTOS** / Bare-metal

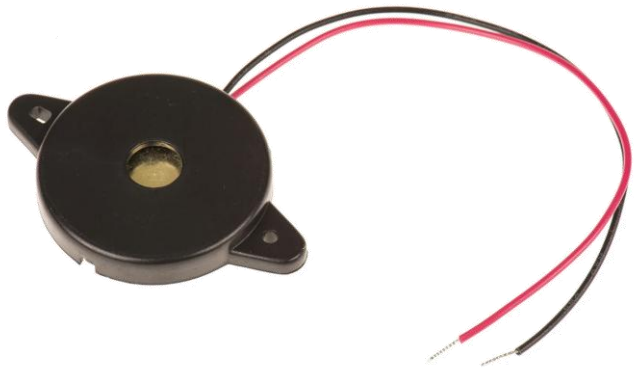


CHERI ports of MicroPython

- initially ported to Morello, funded by Dstl
- later ported to Sonata / CHERIoT, funded by EPSRC ‘capable vms’ project

Morello screenshots

```
root@amarena:~ # file ./micropython.purecap

./micropython.purecap: ELF 64-bit LSB pie executable, ARM aarch64, C64, CheriABI,
    version 1 (SYSV), dynamically linked, interpreter /libexec/ld-elf.so.1, FreeBSD-style,
    stripped

root@amarena:~ # ./micropython.purecap

MicroPython v1.20.0-1182.g4c83449fc.dirty on 2025-02-11; linux [GCC 4.2.1] version

Use Ctrl-D to exit, Ctrl-E for paste mode

>>> print('hello world’)

 hello world
```



CHERIoT GPIO demo

```
def cycle(pin, freq):
    pin.on()
    for i in range(freq):
        ...
    pin.off()
    for i in range(freq):
        ...

def play_note(pin, ltr):
    freq = FREQS[ltr]
    for i in range(50):
        cycle(pin, freq)
        i += freq
```

```
twinkle_twinkle = """
C C G G A A GG
F F E E D D CC
G G F F E E DD
G G F F E E DD
C C G G A A GG
F F E E D D CC
"""

if __name__ == "__main__":
    p = Pin(("rpi", 6), Pin.OUT)
    play(p, twinkle_twinkle)
```


CHERI

versions of MicroPython we support

- CheriBSD/Morello
- Linux/Morello
- CHERIoT-RTOS/Sonata
- baremetal RISC-V/Ibex

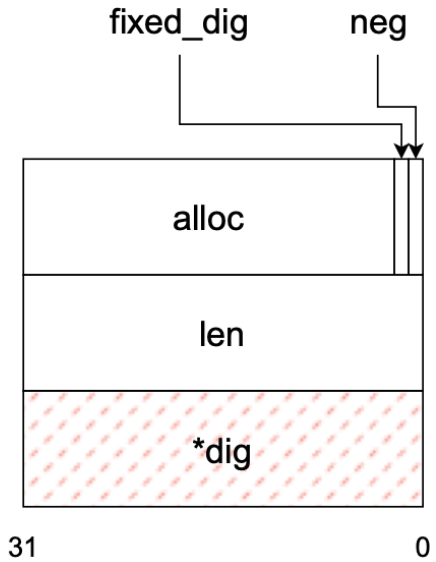
CVE Mitigation

- 5 MicroPython CVEs reported
- 3 are spatial safety violations, which we mitigate directly
- 2 are temporal safety violations, which we don't address
- 60% of MicroPython CVEs mitigated by CHERI

Security Mitigation

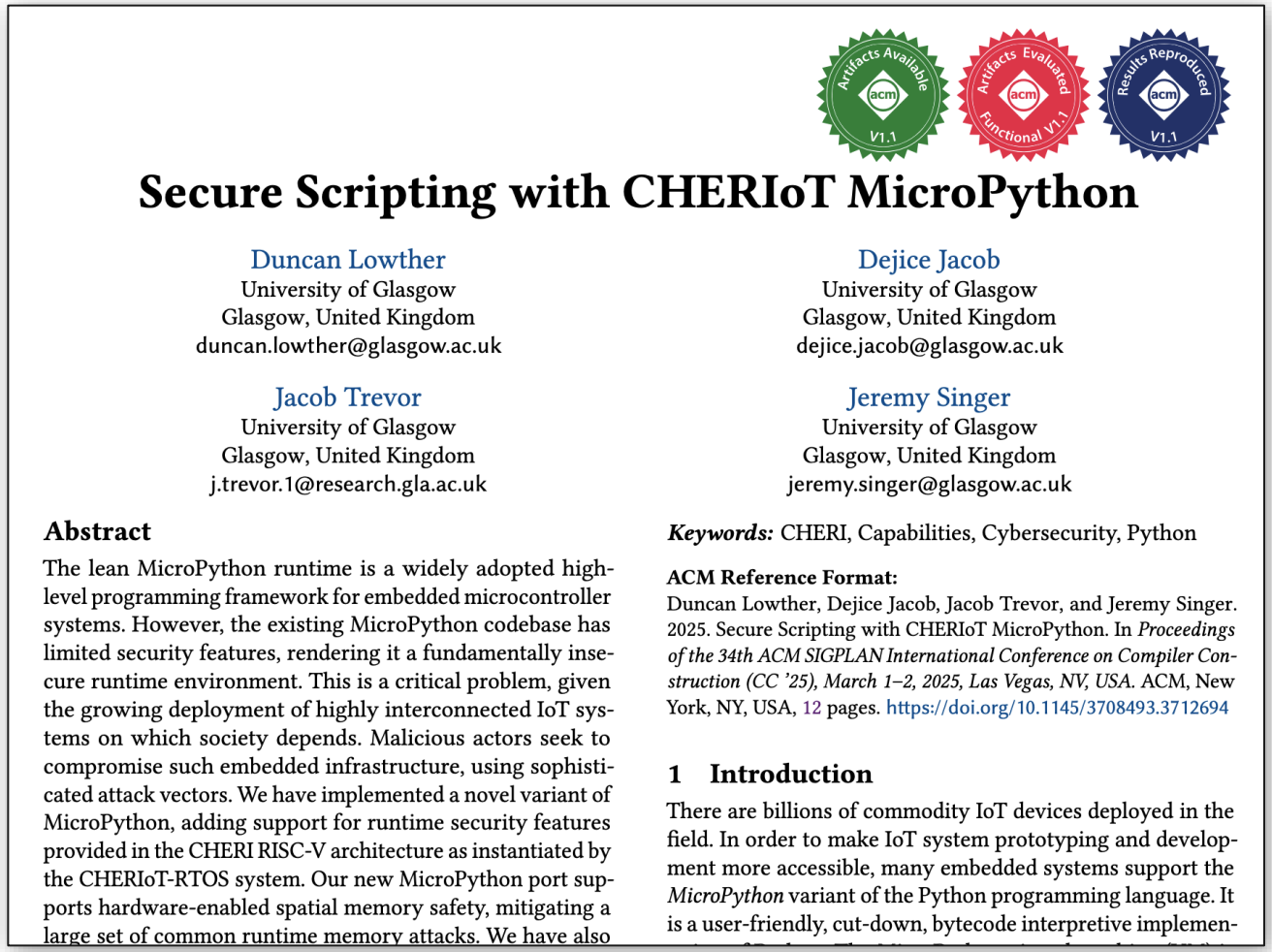
CVE-2023-7158

```
class A:
  def __getitem__(self, idx):
    return idx
print(A()[:].indices(.0))
```




```
mp_int_t mp_obj_int_get_checked(mp_const_obj_t self_in) {
    const mp_obj_int_t *self = MP_OBJ_TO_PTR(self_in);
    mp_int_t value;
    if (mpz_as_int_checked(&self->mpz, &value)) {
    ....

bool mpz_as_int_checked(const mpz_t *i, mp_int_t *value) {
    mp_uint_t val = 0;
    mpz_dig_t *d =  i->dig + i->len ;
```

Check out our CC 2025 paper!



Artifacts Available acm V1.1 · Artifacts Evaluated acm Functional V1.1 · Results Reproduced acm V1.1

Secure Scripting with CHERIoT MicroPython

Duncan Lowther
University of Glasgow
Glasgow, United Kingdom
duncan.lowther@glasgow.ac.uk

Dejice Jacob
University of Glasgow
Glasgow, United Kingdom
dejice.jacob@glasgow.ac.uk

Jacob Trevor
University of Glasgow
Glasgow, United Kingdom
j.trevor.1@research.gla.ac.uk

Jeremy Singer
University of Glasgow
Glasgow, United Kingdom
jeremy.singer@glasgow.ac.uk

Abstract

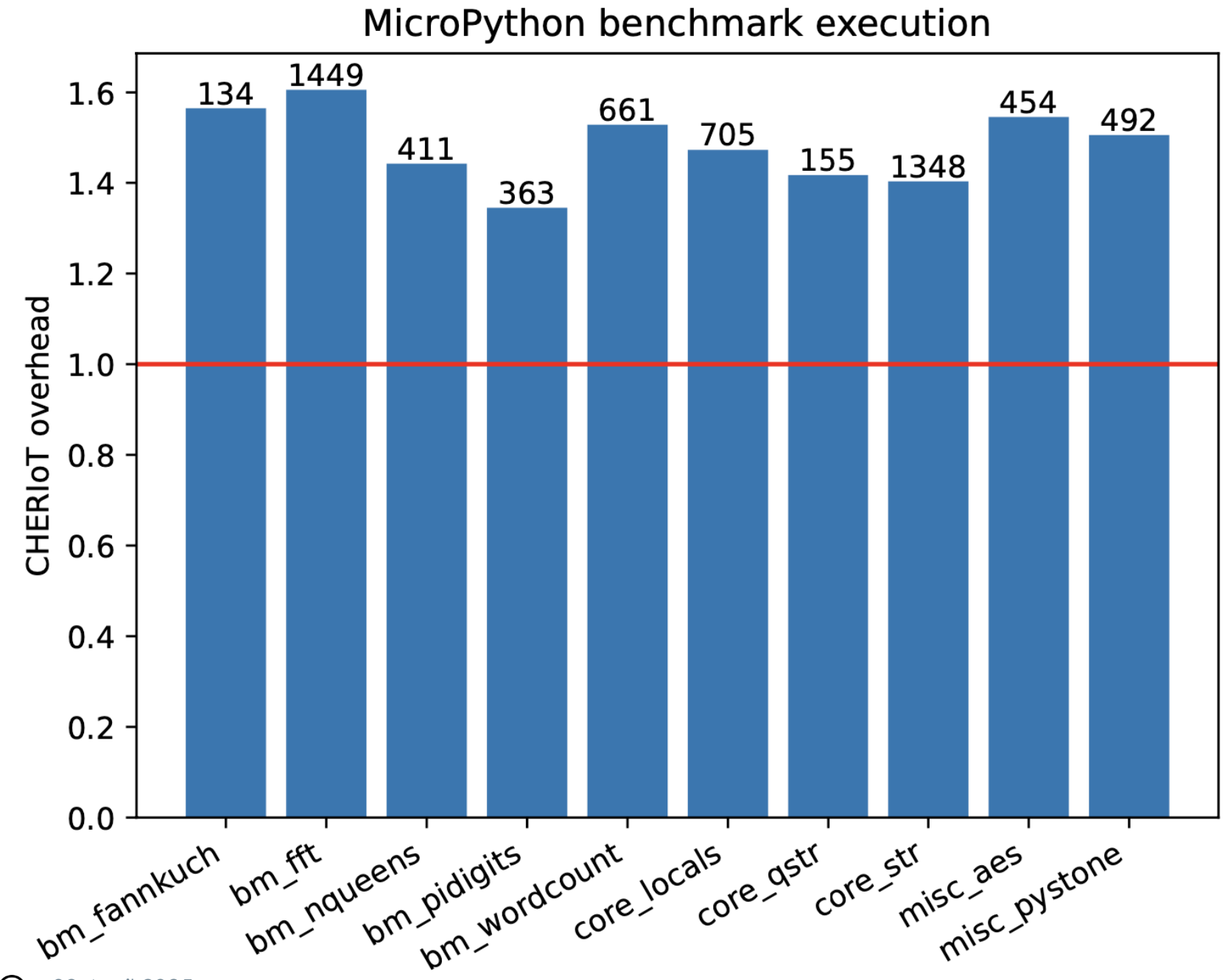
The lean MicroPython runtime is a widely adopted high-level programming framework for embedded microcontroller systems. However, the existing MicroPython codebase has limited security features, rendering it a fundamentally insecure runtime environment. This is a critical problem, given the growing deployment of highly interconnected IoT systems on which society depends. Malicious actors seek to compromise such embedded infrastructure, using sophisticated attack vectors. We have implemented a novel variant of MicroPython, adding support for runtime security features provided in the CHERI RISC-V architecture as instantiated by the CHERIoT-RTOS system. Our new MicroPython port supports hardware-enabled spatial memory safety, mitigating a large set of common runtime memory attacks. We have also

Keywords: CHERI, Capabilities, Cybersecurity, Python

ACM Reference Format:
Duncan Lowther, Dejice Jacob, Jacob Trevor, and Jeremy Singer. 2025. Secure Scripting with CHERIoT MicroPython. In *Proceedings of the 34th ACM SIGPLAN International Conference on Compiler Construction (CC '25), March 1–2, 2025, Las Vegas, NV, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3708493.3712694

1 Introduction

There are billions of commodity IoT devices deployed in the field. In order to make IoT system prototyping and development more accessible, many embedded systems support the *MicroPython* variant of the Python programming language. It is a user-friendly, cut-down, bytecode interpretive implemen-



1.48x overhead

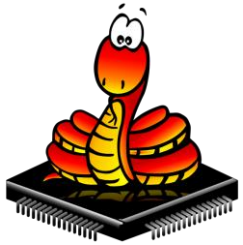
Development Goals

- Update for latest Sonata version
- Run on SCI Iceni
- More Python library support
- Increased compartmentalization

CHERI Virtual Machine Working Group

- CHERI-VM-WG
- focus on CHERI support for virtual machines and managed language runtimes, e.g. Java, Python, JavaScript
- cover both server class (CheriBSD, Linux) and microcontroller class (CHERIoT)

OpenJDK



CHERI

For more details...

- Jeremy.Singer@glasgow.ac.uk

University of Glasgow

CHERIoT–MicroPython repo

https://github.com/glasgowPLI/micropython/tree/cheriot-dev

CHERIoT paper

https://doi.org/10.1145/3708493.3712694