# Final Year Project: Michael (DSbD)

Draft title: Applicability of DSbD to performance-sensitive networking applications

## Background

Digital Security by Design is a UK government backed transformative initiative to create a secure foundation for future computing.

The technology underlying the DSbD initiative is CHERI (Capability Hardware Enhanced RISC Instructions), developed at the University of Cambridge and SRI. CHERI is a set of architecture enhancements applied to CPU instruction sets which enable fine-grained memory access protections (read/write/execute permissions, bounds checking, memory compartmentalisation within a process)

Morello is an ARM-led research programme which includes a prototype architecture incorporating CHERI concepts, as well as a development board ('the Morello board') containing an SoC which implements this prototype architecture.

CheriBSD is an adaptation of the FreeBSD operating system to support Morello which was developed at the University of Cambridge.

Data Path Development Kit (DPDK) is an open source packet processing library, originally developed by Intel, and widely used today in networking and storage applications and appliances.

## Project Description

The area of investigation is the application of DSbD technologies to performance sensitive distribution of data from a broker to untrusted consumer plugins. A concrete example of this is in a networking application or appliance (such as a firewall or network monitor) which receives packets from an incoming network device, classifies the packets and distributes them to one or more '3rd-party' untrusted plugins for consumption.

Such a program structure entails an undesirable trade-off between performance (co-located plugins within the main address space rather than separate plugin processes with consequent IPC overhead) and security (isolated address spaces for untrusted code vs their inclusion in the address space of the main process, which may allow them unauthorised access to additional data)

We would like to demonstrate that DSbD can permit elimination of this trade-off and enable an architecture which is both **secure** and **performant**.

* Develop packet processing application for CheriBSD and execute this application on a Morello board.
* Classify incoming packets and dispatch them to consumers
* Design and implement CHERI-enabled in-process plugins (“DSbD design”) and traditional multi-process structure
* Add protection to DSbD design (compartmentalisation of packet buffers based on target consumer, bounds checking and appropriate permissions) and demonstrate that it works (**secure**)
* Evaluate performance in both architectures and demonstrate that DSbD design is more **performant** or if not, identify and describe why this is the case
* Write up project and conclusions

The project area is related to investigative activity Pytilia has previously undertaken, but the project described here is a new and distinct piece of work. It is also unrelated to the work Michael carried out during his placement year with Pytilia.

## Objectives

1. Familiarisation with Morello, CheriBSD and CHERI concepts
2. Successfully boot, install and configure CheriBSD on the supplied Morello board
3. Develop a simple packet processing application in both IPC and single-process, CHERI-enabled varieties. Use of DPDK is suggested but at the student’s discretion. The CHERI-enabled application should support bounds checking, permissions enforcement and memory compartmentalisation.
4. Define simple packet streams (including a range of packet sizes) and transmit those streams to both varieties of the packet processing application.
5. Measure and analyse key performance characteristics including packet processing latency and CPU utilisation for both varieties and write these up in a final report.

## Learning Outcomes

1. Hands-on knowledge of CHERI including CheriBSD
2. Enhanced C and systems/embedded programming capability
3. Ability to identify, measure, record and interpret key performance metrics

## Relevant Skillset

* Embedded systems
* C/C++ programming
* Networking/packet processing
* Computer Architecture/Operating Systems
* Cyber-security including secure hardware

## Further Reading

* [University of Cambridge CHERI](https://www.cl.cam.ac.uk/research/security/ctsrd/cheri/)
* <https://www.dsbd.tech/>
* <https://www.arm.com/architecture/cpu/morello>
* [Security Analysis of CHERI (Microsoft)](https://github.com/microsoft/MSRC-Security-Research/raw/master/papers/2020/Security%20analysis%20of%20CHERI%20ISA.pdf)