Electrical & Electronic Engineering, Software & Electronic Systems Engineering
Final Year Projects 2022-2023

Applicability of DSbD to Performance-Sensitive Networking Applications

Supervisor: Sandra Scott-Hayward

	Control	x	Embedded Systems		High Frequency Electronics		Microelectronics
	Electric Power	x	Software		Connected Health		MEMS
x	Cyber-Security	x	Wireless Communications		Signal/Image Processing		Intelligent Systems
	Digital Design		Sensor Networks	x	Data Analytics		Electronics

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.

Objectives

1. Familiarisation with Morello, CheriBSD, and CHERI concepts.
2. Successfully boot, install, and configure CheriBSD on the supplied Morello board.
3. Develop a packet processing application for CheriBSD and execute this application on a Morello board.
4. Classify incoming packets and dispatch them to consumers.
5. Design and implement CHERI-enabled in-process plugins ("DSbD design") and traditional multi-process structure.
6. Add protection to DSbD design (compartmentalisation of packet buffers based on target consumer, bounds checking and appropriate permissions) and demonstrate that it works (secure).
7. Define simple packet streams (including a range of packet sizes) and transmit those streams to both varieties of the packet processing application.
8. Measure and analyse key performance characteristics including packet processing latency and CPU utilisation for both varieties and write these up in a final report.
9. Write up project and conclusions.

Learning Outcomes

Upon completion of the project you will expect to have:

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.