QUB Presentation Plan

Assessment Criteria:

* Demonstration (33%)
  + Demonstrate the system very well
* Contribution Analysis (34%)
  + Breadth of contextual analysis
  + Differentiation in this project compared to existing solutions
* Contribution Depth (33%)
  + Understanding of the system
  + Understanding of other approaches
  + Problem
    - Memory safety issues causing security vulnerabilities
    - C/C++ rely on programmer competence and honesty
    - Pointers only store an address
    - Buffer overflows are the most notorious of these issues
    - Many applications exist which use C and/or C++
    - Extra code needs to be generated for security
      * Performance penalty

**Content**

Introduction:

* Title slide
* Project overview
  + Construct a packet processing application which reads in packets
  + Classifies and dispatches the packets to one or more consumers
  + Two types of consumers
    - Consumers in separate process/address space as different applications, interaction via IPC
    - Consumers in the same process/address space as plugins to the packet processing application, interaction via DSbD technologies
  + Show the CHERI consumers are safe and secure
  + Evaluate the performance metrics
    - Packet processing latency
    - Total CPU utilisation

DSbD:

* Overview
  + An initiative which fundamentally reworks the computing architecture in a more secure way
  + A joint research project by the University of Cambridge and SRI International
  + The underlying technology is CHERI
    - Architectural extension to RISC ISAs
    - Replaces pointers with capabilities
    - More to follow
* CHERI Architectural Capabilities
  + Write about capabilities
    - Section 2.2 - <https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-941.pdf>
    - Section 2 - <https://www.cl.cam.ac.uk/research/security/ctsrd/pdfs/201904-asplos-cheriabi.pdf>
  + Diagram?
* Morello Board
  + ARM has developed a custom architecture called the Morello board with capability support
  + Runs with a capability-enabled OS called CheriBSD
  + Picture of Morello board

Project Motivation:

* Firewall
  + Find details
* Network Monitor
  + Find details

Packet Processing Application:

* DPDK was ported to the Morello board
  + Commonly used framework in industry
  + Focused on providing high-speed packet processing
    - Lines up with the project specification
* A
* A
* A
* A
* CHERI-enabled mode
  + Details
* IPC-enabled mode
  + Details
* No processing mode
  + Details

CHERI-based Security:

* Bounds
  + Tightened to match packet length
  + Tested by attempting to read one character beyond the capability bounds
  + Correct error raised
  + Screenshot of capability bounds error
* Permissions
  + Restricted to read-only
  + Tested by attempting to overwrite one character
    - Valid character location within capability bounds
  + Correct error raised
  + Screenshot of capability permission error

Testing Procedure:

* Quiet printing mode used to reduce the noise in the results
* Ran for varying packet streams
  + Variable packet counts
  + Variable packet sizes
* Averaged over 10 iterations for each measurement
* Identical metrics recorded for CHERI and IPC methods of security, as well as no processing mode

Results:

* Packet Processing Latency (Slide 1)
  + Packet count graphs
    - Unadjusted & adjusted
  + Explain writing to buffer took time
  + Explain adjusted graph derivation
  + Both modes of operation take constant time as packet count increases
  + CHERI has a significantly shorter latency
* Packet Processing Latency (Slide 2)
  + Packet size graphs
    - Unadjusted & adjusted
  + Same graph process
  + IPC increases latency as packet size increases
  + CHERI takes a constant time
  + Makes sense because IPC passes by value and CHERI passes by pointer (equivalent)
* CPU Utilisation
  + Graphs
    - Packet count and size
  + CPU utilisation increases as packet size increases
  + IPC has more variation in results

Conclusions:

* Successfully completed the project objectives
* CHERI is secure and fit for use
* CHERI removes the need for added code for security
* CHERI outperforms the IPC method in both packet processing latency and CPU utilisation
* CHERI effectively allows a secure way to pass by reference instead of pass by value, giving a lot of potential for performance

Demonstrations:

* CHERI & IPC modes of operation
  + Use normal printing
* CHERI & IPC modes of operation
  + Use quiet mode
* Capability bounds error
* Capability permissions error

Structure:

* Title slide/personal introduction
* Introduction
  + Packet processing
  + Firewall
  + Network monitor
* DSbD
  + CHERI
  + Morello
  + CheriBSD
* Existing Work
  + Pytilia created a packet processing application for Morello FVP
  + Used DPDK
  + Port not compatible with physical Morello board
  + No performance metrics for viability
* Project
  + Create a packet processing application on Morello board
  + Port DPDK for use
  + Operate in two modes, CHERI-based security and IPC
  + Validate CHERI security (bounds and permissions)
  + Record performance metrics for both modes of operation
  + Compare for application viability
* Setup
  + Morello board
    - Install CheriBSD on Morello board
    - Set up network connectivity
      * Ethernet
      * DHCP reservation
      * SSH into board externally
    - Picture
  + Port DPDK
    - Start from Pytilia Morello FVP branch
    - Meson build
    - Ninja build
    - Helloworld
    - Limitations
      * One process
* Packet processing application
  + Based on previous Pytilia application
  + Receives packets from pcap
  + Classifies packets according to packet data
  + Sends packets to corresponding consumer
  + Consumer increments a counter upon receiving the packet
  + After all packets sent, consumers send counters to application
  + Total counters displayed
  + Total packet processing time displayed
* Modes of operation
  + Single process, CHERI-based security
  + IPC, separate application, UDP sockets
* Single process
  + Consumer in a plugin
  + Plugin runs in same address space and process as packet processing application
  + Secured by CHERI capabilities
    - Bounds tightened to packet length
    - Write permissions removed
  + Calls to consumers to retrieve counters
  + Different from Limelight
    - Previously, buffer validated
    - Now, derived capability for packet validated
    - Allows buffers to be reused
* IPC
  + Consumer in an external application
  + Plugin runs in different address space and process
  + Packets sent through UDP sockets
  + Finished packet sent after last packet is sent
  + Consumers send counters through UDP
* Testing
  + Packet generation
    - Python script made to generate packets
    - Table of packet types
    - Input for packet processing application
* Performance tests
  + Both modes of operation ran with each packet stream 10 times
  + Packet processing latency and total CPU utilisation recorded by Python script
  + Stored in a csv
* Initial Results
  + Packet count
    - CHERI faster
    - Less CPU utlisation
  + Packet size
    - CHERI faster
    - Less CPU utilisation
    - Increase in packet processing latency odd
      * Uncovered discrepancy in buffer read time
* No processing mode
  + Reads in packets to buffer
  + Clears buffer
  + No processing
  + Performance tested
    - Removes noise from results
    - Metrics recorded in separate runs to recorded results
* Final results
  + Tests reran
  + Adjusted packet count same as before
  + Adjusted packet size
    - CHERI speed is independent of packet size
    - Makes sense, effectively passing by pointer instead of passing by value
* Conclusions
  + DPDK successfully ported to Morello board
  + CHERI is secure for packet processing applications in same plugin
  + CHERI is proven to be faster and use less CPU
  + Viable for use in applications
* Questions?

Structure:

* Title slide/personal introduction
* Background
  + Networking applications
* DSbD
  + CHERI
  + Morello
  + CheriBSD
* Existing Work
  + Pytilia created a packet processing application for Morello FVP
  + Used DPDK
  + Port not compatible with physical Morello board
  + No performance metrics for viability
* Project
  + Create a packet processing application on Morello board
  + Port DPDK for use
  + Operate in two modes, CHERI-based security and IPC
  + Validate CHERI security (bounds and permissions)
  + Record performance metrics for both modes of operation
  + Compare for application viability
* Setup
  + Morello board
    - Install CheriBSD on Morello board
    - Set up network connectivity
      * Ethernet
      * DHCP reservation
      * SSH into board externally
    - Picture
  + Port DPDK
    - Start from Pytilia Morello FVP branch
    - Meson build
    - Ninja build
    - Helloworld
    - Limitations
      * One process
* Packet processing application
  + Based on previous Pytilia application
  + Receives packets from pcap
  + Classifies packets according to packet data
  + Sends packets to corresponding consumer
  + Consumer increments a counter upon receiving the packet
  + After all packets sent, consumers send counters to application
  + Total counters displayed
  + Total packet processing time displayed
* Modes of operation
  + Single process, CHERI-based security
  + IPC, separate application, UDP sockets
* Single process
  + Consumer in a plugin
  + Plugin runs in same address space and process as packet processing application
  + Secured by CHERI capabilities
    - Bounds tightened to packet length
    - Write permissions removed
  + Calls to consumers to retrieve counters
  + Different from Limelight
    - Previously, buffer validated
    - Now, derived capability for packet validated
    - Allows buffers to be reused
* IPC
  + Consumer in an external application
  + Plugin runs in different address space and process
  + Packets sent through UDP sockets
  + Finished packet sent after last packet is sent
  + Consumers send counters through UDP
* Testing
  + Packet generation
    - Python script made to generate packets
    - Table of packet types
    - Input for packet processing application
* Performance tests
  + Both modes of operation ran with each packet stream 10 times
  + Packet processing latency and total CPU utilisation recorded by Python script
  + Stored in a csv
* Initial Results
  + Packet count
    - CHERI faster
    - Less CPU utilisation
  + Packet size
    - CHERI faster
    - Less CPU utilisation
    - Increase in packet processing latency odd
      * Uncovered discrepancy in buffer read time
* No processing mode
  + Reads in packets to buffer
  + Clears buffer
  + No processing
  + Performance tested
    - Removes noise from results
    - Metrics recorded in separate runs to recorded results
* Final results
  + Tests reran
  + Adjusted packet count same as before
  + Adjusted packet size
    - CHERI speed is independent of packet size
    - Makes sense, effectively passing by pointer instead of passing by value
* Conclusions
  + DPDK successfully ported to Morello board
  + CHERI is secure for packet processing applications in same plugin
  + CHERI is proven to be faster and use less CPU
  + Viable for use in applications
* Questions?