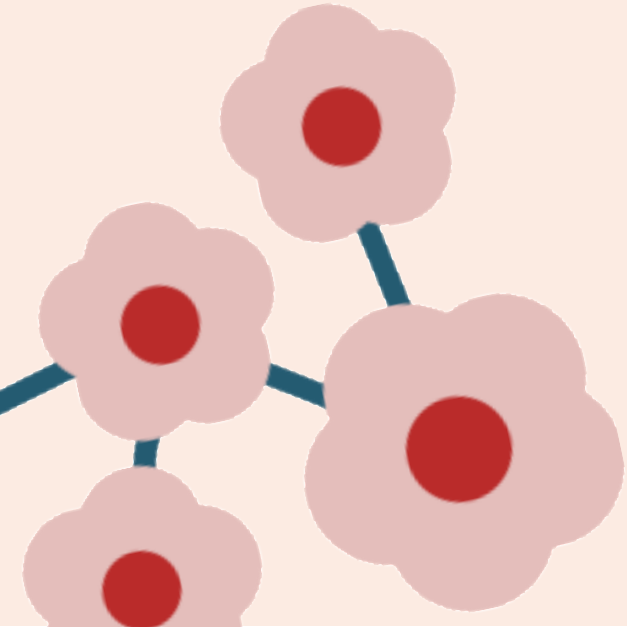
09 April 2025



Securing Secure Networking tools for Morello Linux

Joshua Lant- joshua.lant@manchester.ac.uk
Research Associate- University of Manchester

Morello at the Edge (MoatE)

The University of Manchester

- Built commercial demonstrator system for edge computing.
- Morello acting as a secure server.
- Linux-Morello w/ Yocto Distribution.
- UoM- Enable secure networking:
  - VPN (WireGuard)
  - Firewall/Packet Filtering (Netfilter/nftables)

UK Research and Innovation

Digital Security by Design

ICEOTOPE COOLING REDEFINED

The University of Manchester

THE GOOD PENGUIN DEPENDABLE SOFTWARE SOLUTIONS

edge synergies

WireGuard VPN

- Minimal porting effort for WireGuard itself.
  - In-kernel implementation
  - We do not deal with other implementations.
  - Only userspace configuration tools to port.
- Many configurations and tests use Netfilter.
  - Forwarding between interfaces (namespaces)
  - NAT for use as VPN gateway
  - Stateful firewall rules (conntrack/rate limiting)

nftables and Netfilter Subsystem

- De-facto standard firewall tool for Linux
- nft/iptables-nft – userspace tool for configuration
  - libnftnl- netlink programming API
    - libmnl- parsing/building netlink packets
- Netfilter is a large subsystem, highly configurable.
  - Many vulnerabilities:
    - Use-after-free, out of bounds access
    - High severity, privilege escalation
    - Several publicly posted, proven exploits (CVE-2024-1086, CVE-2023-32233, CVE-2022-1015)

CHERI

Issues with porting

- Kernel pointers in Netfilter's UAPI headers
  - Structs incorrectly sized by userspace
  - Netlink expressions padded incorrectly
  - New type required for netfilter headers

```
#ifdef CONFIG_CHERI_PURECAP_UABI
    typedef __uintcap_t __nf_kptr_t;
#else
    typedef unsigned long __nf_kptr_t;
#endif
```

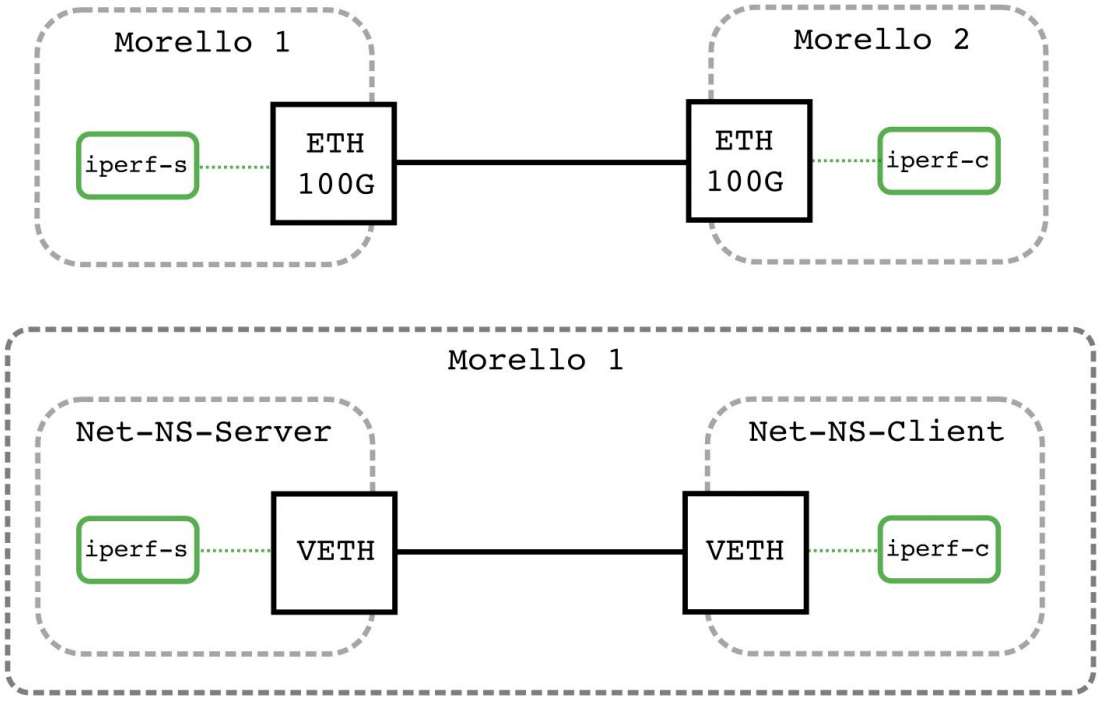
```
struct xt_entry_match {
…
    struct xt_match *match;
…
```

```
struct xt_entry_match {
…
    union {
        struct xt_match *match;
        __nf_kptr_t __match;
    };
…
```

Issues with porting

- Cached kernel headers in userspace tools
  - Broken build using custom kernel header locations
    - General lack of maintenance for non-standard build configs.
- Issues building against musl-libc
- Kernel Selftests broken by purecap
  - Scripts dependent on many userspace packages.
    - Specific versions/functionality
- Several persistent ip/nftables test failures
  - time consuming to diagnose

Network Performance



- Intel e810qc 100G cards installed
  - 40-45G through copper links
    - Iperf3, -P = 2
    - MTU has no effect (offload)
  - 80G baseline w/ namespaces
  - C64 == aarch64 performance
- WireGuard VPN
  - ≈11% latency overhead from purecap, with significantly less jitter
  - 2.5G for a single peer
  - Large overheads seen using network namespaces?
- No throughput overhead from purecap iptables-nft

Open Questions

The University of Manchester

- Protections beyond traditional network security?
  - What more needs to be done in-kernel?
- Wish to test on CHERI-Linux
  - Safety in the kernel
  - Additional temporal safety needed?
    - Netfilter: many use-after-free vulnerabilities
- What does purecap kernel mean for porting effort?
  - Discrepancy between user/kernel space removed
  - New challenges created, dependencies, device drivers?

Questions?

- Please find me to discuss more…

- Kernel patches available.
- User packages in open-source Yocto layer soon
  - minor modifications needed.

- Also… ask me about our other MoatE work; porting the MAMBO Dynamic Binary Modification tool.