Remotely Controlling TrustZone Applications? A Study on Securely and Resiliently Receiving Remote Commands

Shengye Wan, Kun Sun, Ning Zhang, Yue Li

William & Mary
George Mason University
Washington University in St. Louis

June 29, 2021

OUTLINE

- Introduction
- System Overview
- System Evaluation
- Takeaways

INTRODUCTION

Background: Mobile Device Management

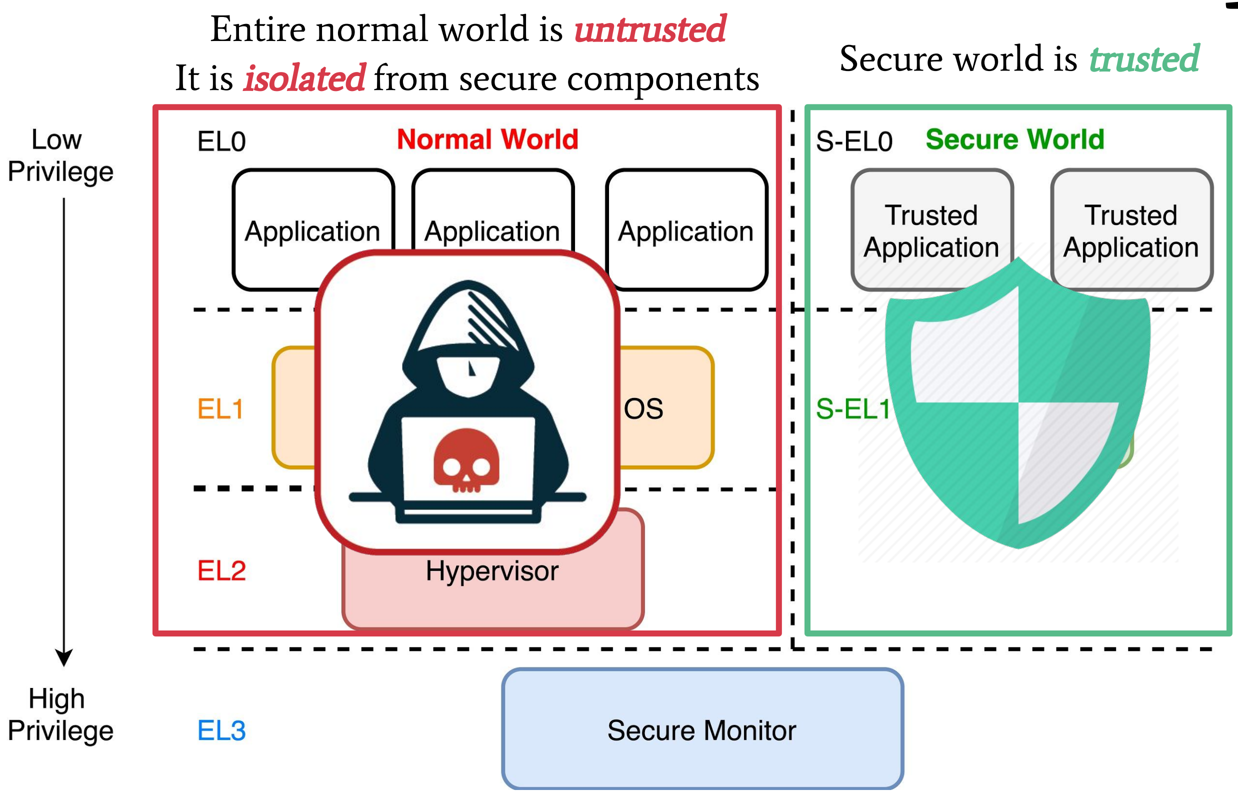
- Mobile device management (MDM)
  - Enable corporate administrators to remotely perform essential functions
    - Supportability, security, and corporate functionality

Security of MDM Agents

- MDM workflow
  - Administrator <-> Management Commands <-> MDM Agents (clients)
- MDM agents are security-sensitive
  - Rich OS cannot be trusted to hold MDM agents
    - 859 CVEs are reported in 2020 for Android [1]
  - Opportunities to enhance MDM agents' security

[1] CVE Details: https://www.cvedetails.com/vulnerability-list/vendor_id-1224/product_id-19997/year-2020/Google-Android.html

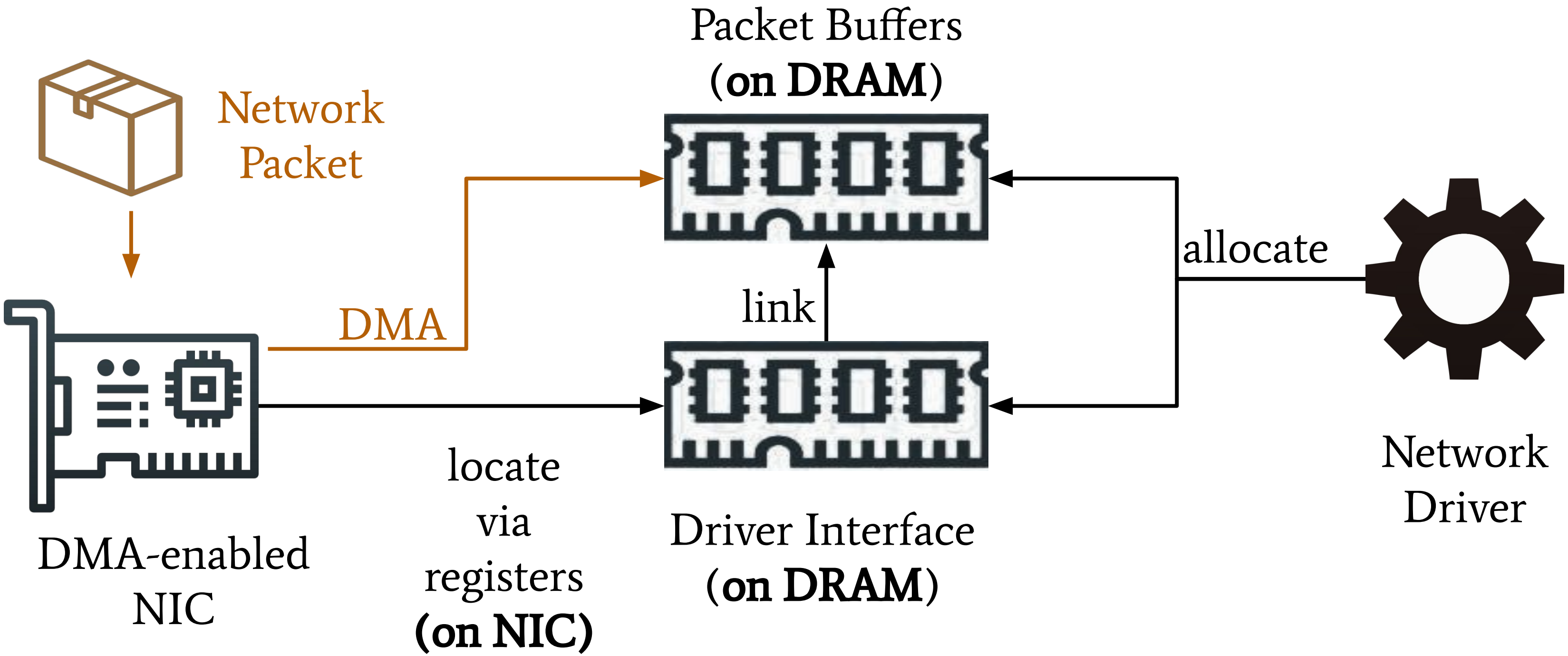
Background: ARM TrustZone Technology



Motivation: Two Worlds Need to Share One NIC

- MDM agents require network service
  - Remote attestation, remote control, remote troubleshooting
- Secure world (SW) does not have an exclusive NIC
  - Commercial devices only equip one set of network devices
    - Limited hardware spaces on mobile
  - NW and SW need to share the NIC
- **Question:** With a shared Network Interface Card (NIC), how to provide a reliable network for ARM TrustZone secure world?

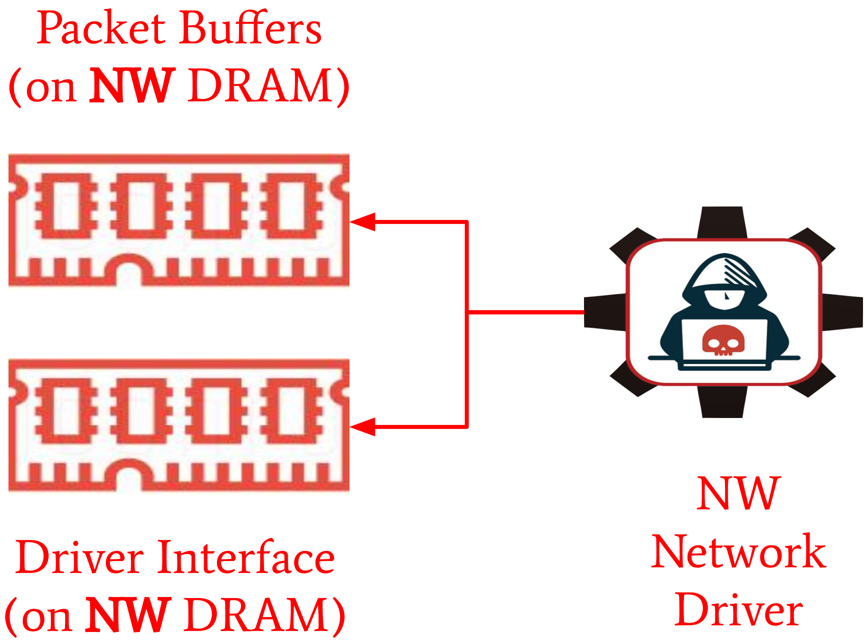
Background: NIC Workflow



SYSTEM OVERVIEW

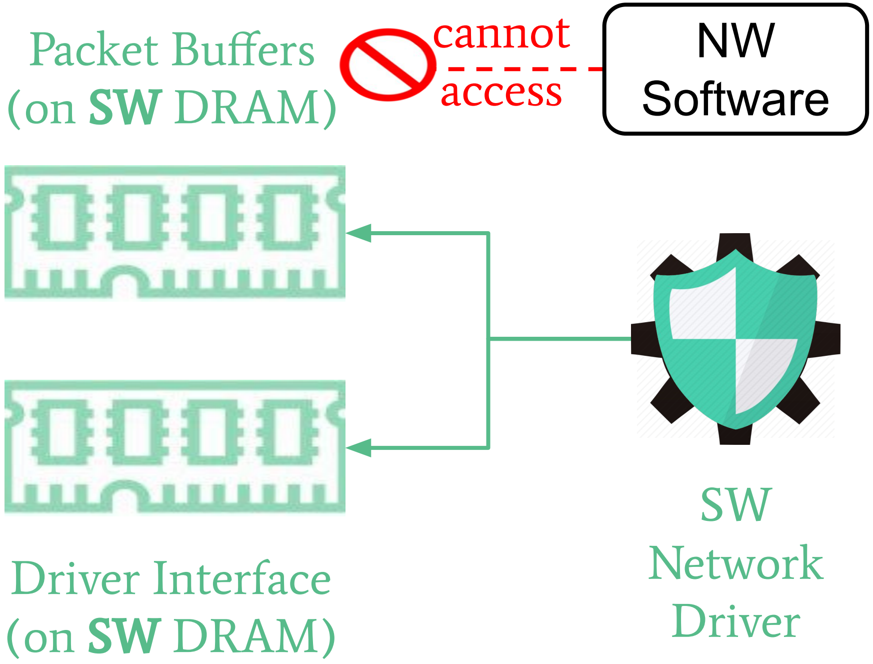
How to Share One NIC Between Two Worlds?

- Option-1: sharing the single network driver in NW
    - Pros: providing good normal world performance
    - Cons: not reliable for the secure world



Sharing One NIC: Option-2

- Option-2: sharing the single network driver in SW
  - Pros: reliable for SW
  - Cons: introducing large overhead
    - NW software cannot access packet buffers directly

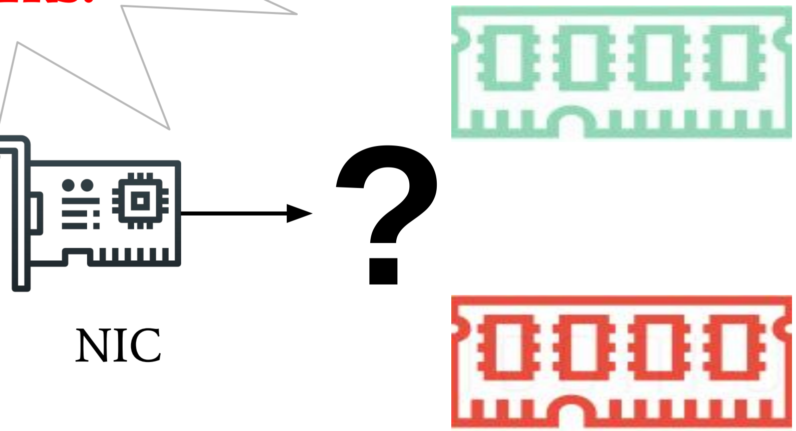


Sharing One NIC: Option-3

- Option-3: depl… ing t… n… orl… rivers in each world
  - Pros: reliable an… performance
  - Cons: very difficult to sch… dule tw… drivers
    - One NIC only connects to one driver's interface

None of these options works!

…W Driver Interface + Packet Buffers

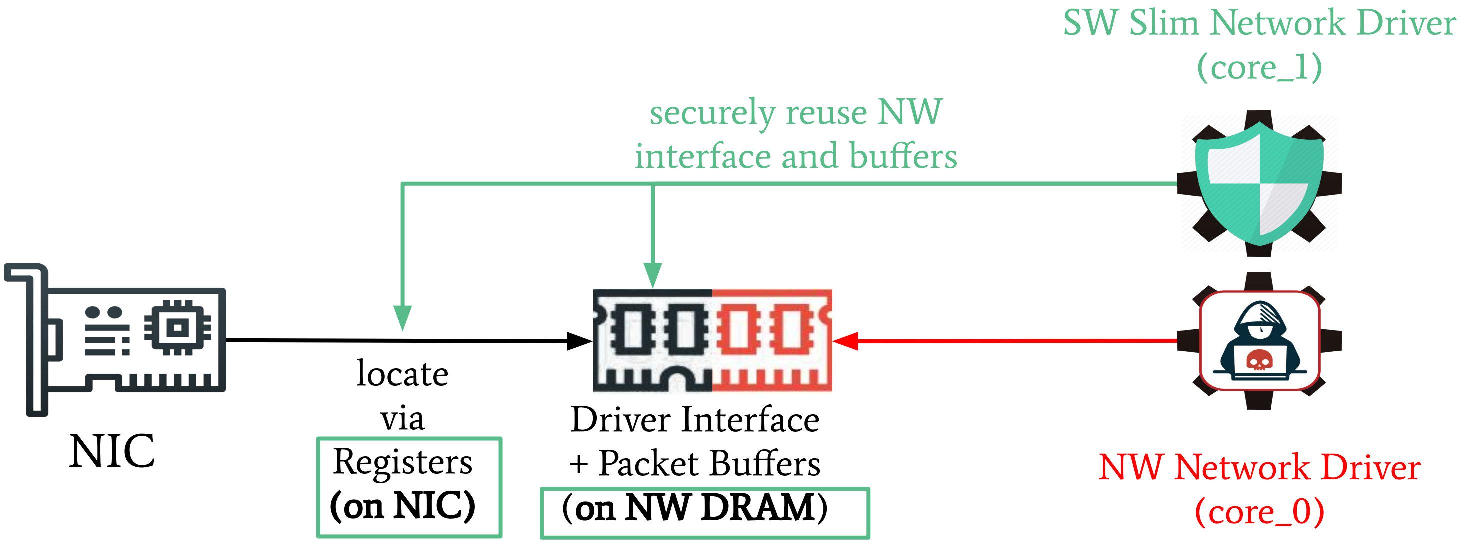


NIC

NW Driver Interface + Packet Buffers

Our Solution: TZNIC

- Deploying a complete NW-driver and a slim SW-driver
  - **Key idea:** executing two drivers simultaneously on the multi-core platform
  - Multiplexing the NW-driver's interface



TZNIC Challenges

1. Filling the semantic gap to use NW-driver's interface reliably
   - SW-driver should not put any trust in the normal world
   - SW-driver should not require any collaboration from the normal world

2. Resisting interference from the normal world
   - Securely sharing the interface and buffers with NW-driver

Resolving Challenge-1: Filling Semantic Gap

- Locating NW driver's interface via the NIC registers
  - Registers indicate the ring buffer information
  - Registers are readable to the secure world
- Locating the packets via the NW driver's interface
  - Interface and buffers are saved in the NW memory
    - Secure world has the privilege to read/write
  - NW driver uses fixed-format interface to communicate with NIC
- Does not request any collaboration of the normal world

Resolving Challenge-2: Resisting NW interference

- Reading packets in parallel of NW-driver
  - SW-driver wakes itself periodically to receive the packets
  - One receiving buffer can be read by two drivers simultaneously

- Saving the secure-world packets to the secure memory
  - Each buffer should be independent and loss-tolerant (e.g., UDP)
  - Normal-world attacker cannot access

SYSTEM EVALUATION

TZNIC Implementation

- Implementing our prototype based on ARM-TF [2]
  - Marvell Yukon-II NIC & Marvell sky-2 driver (v 1.30)

- TZNIC's slim driver's size is 18.63% of the original driver
  - Full-fledged normal-world sky-2 driver: 5707 LOC
  - TZNIC slim secure-world driver: 1063 LOC

[2] ARM-software. ARM-Trusted-Firmware. GitHub.

TZNIC Evaluation - Reliability

- Attacker capacity
  - Brute-force deleting the packet from a specific IP
  - Benchmark iPerf [3] cannot receive any packet under our interference

- Under the interference of our attacker
    - TZNIC receives 67% of the packets on average
      - 22% - 92%

[3] Dugan et al.. iPerf Benchmark. https://iperf.fr

TAKEAWAYS

Summary

1. We can support software in TrustZone secure world with reliable network
2. Secure-world driver can reliably reuse the normal-world driver's interface
    a. Secure world has higher privilege to inspect on-device registers
    b. Secure world has higher privilege to read normal-world driver's data
    c. Secure world has higher privilege to get activated
3. TZNIC makes 0 modifications or requirements on the rich OS

Thanks & Questions?

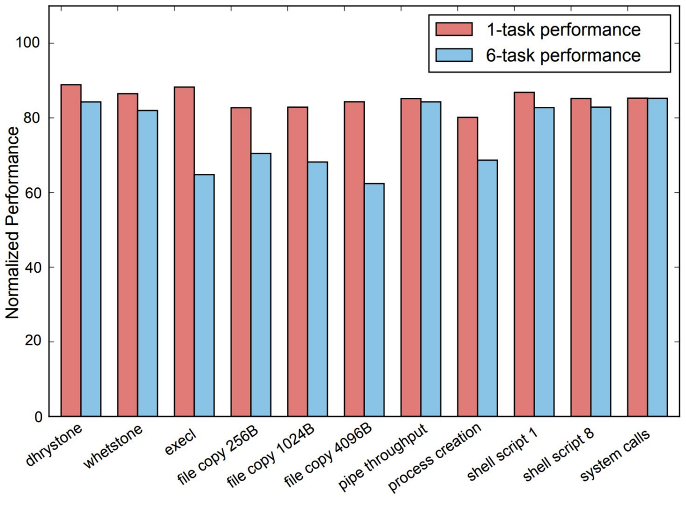
Shengye Wan

June 29, 2021

Backup Slides

TZNIC Evaluation - Rich OS Overhead

- When TZNIC wakes up, rich OS will suffer 16.7% overhead
- The overall overhead can easily improved
    - TZNIC does not wake up often
        - The wake-up frequency can be adjusted
    - To promise 95% of the rich OS performance:
        - TZNIC wakes 10ms among every 80ms



Future Works

1. Protecting network devices from Denial-of-Service attacks
    - Configuring the NIC as a secure-world hardware
2. Deploying multiple TZNIC in secure world
    - Solution-1: moving TZNIC into secure application layer
    - Solution-2: Using new ARM TrustZone feature
        - Achieve virtualization in the secure world

Background: Cross-World Context Switch

- SMC
  - ARM special instruction to enter the Secure Monitor (EL3) code
  - *Core-i* can only use SMC to switch the status of *core-i*
- Interrupt
  - SW-interrupt is promised to route to secure world
    - Interrupt untrusted NW execution
    - One interrupt may arrive on
      - One specific *core-i* (Private Peripheral Interrupt)
      - Multiple cores (Shared Peripheral Interrupt, Software Generated Interrupts)
  - NW-interrupt can get handled in both worlds