NetReach: Guaranteed Network Availability and Reachability to enable Resilient Networks for Embedded Systems

Tom Van Eyck
DistriNet, KU Leuven
Leuven, Belgium
tom.vaneyck@kuleuven.be

Sam Michiels
DistriNet, KU Leuven
Leuven, Belgium
sam.michiels@kuleuven.be

Xiaojiang Du
Stevens Institute of Technology
Hoboken, NJ, U.S.A.
xdu16@stevens.edu

Danny Hughes
DistriNet, KU Leuven
Leuven, Belgium
danny.hughes@kuleuven.be

***Abstract*—Networked embedded devices are increasingly deployed in safety critical environments such as robotics, smart manufacturing and autonomous vehicles. Availability is an essential prerequisite of safety critical systems, which depend upon timely access to sensed data to inform the real-time control of actuators. Recent work has demonstrated that trusted computing features can be used to guarantee the availability of local resources to the safety-critical applications. However, prior work fails to guarantee the availability of a network connection, which is essential for correct system operation. To address this issue, we contribute NetReach, which uses Arm TrustZone to guarantee network availability to, and the reachability of, critical applications via a secure backup channel. Evaluation of NetReach shows that it can preserve the network connectivity of critical applications while under attack, with a worst case overhead of 18.66 % for networked software running in the Normal World. Furthermore, NetReach introduces minimal additional code in the Secure World (only 418 lines of code). The presented features of NetReach enable future work toward resilient networks.**

***Index Terms*—Arm TrustZone, Network Availability, Network Reachability**

1. Introduction

Industry 4.0 improves efficiency and product quality by connecting industrial appliances to data analytics via a variety of wired and wireless networks. Recent hardware advances have delivered compact and low-cost System on Chips (SoCs) that are capable of running commodity Operating Systems (OS) and networking stacks. These platforms are very attractive to industrial players who use them to connect large numbers of globally distributed devices, for which it is (financially) infeasible to perform manual interventions.

While networked embedded systems enable reconfiguration and restoration of devices, this opens a new attack surface to adversaries who may use industrial appliances to cause significant financial and physical damage. This attack surface is very large: not only are internet-connected applications possible targets, but also the network driver and the wider operating system, which may comprise hundreds of thousands of lines of code, which are often poorly isolated from critical control software in the case of embedded systems.

Perhaps the most common attacks on such industrial networks are Denial of Service (DoS) attacks, wherein the attacker causes outages by making essential devices unresponsive, from within those devices themselves. This denies the availability of the system, which might have very dangerous consequences, even loss of life.

Recently, Arm TrustZone has been proposed as a solution to guarantee the availability of local resources to critical control code [1], [2]. By extending TrustZone with a minimal real-time scheduler and moving critical control code into the Secure World (SW), availability can be guaranteed, even in cases where an attacker gains full control over the Normal World (NW). However, as the network stack runs in the NW, it is trivial for such an attacker to eliminate network connectivity, thereby rendering a device unresponsive to remote management and control and therefore necessitating a complex and costly on-site intervention. Previous work is able to share the network between worlds [3]–[6], but none address this attack vector.

NetReach[1] is the first step toward providing resilient [7] networks that prevent this scenario. It contributes (i.) an **always available** network peripheral in the SW and (ii.) an **always reachable** backup network connection to critical applications, *anticipating* and *withstanding* strong attackers in the NW. Considered as a whole, these features are a significant improvement to the resiliency of networked embedded systems, while minimally increasing the size of the Trusted Computing Base (TCB) by 418 lines of code.

2. Requirements and Assumptions

NetReach aims to preserve network access in the face of a powerful attacker who has control over the NW, and bugs in both user and kernel-space. The following requirements achieve this goal:

1) Protect the network peripheral from being disabled by software running in the NW.
2) Provide a backup network connection to critical applications in case the NW network stack has been compromised.
3) Minimize TCB size to reduce the attack surface.

We assume all hardware is trusted, bug free and protected from physical attacks. In particular, we do

[1] Source code and documentation can be found at: https://gitlab.com/distrinet-netreach

not consider security vulnerabilities of the networking peripheral's hardware features or it's processing firmware. Additionally, we assume all software in the SW is well-tested and free of bugs and vulnerabilities. NetReach has no expectations of, or trust in software running in the NW, which may contain bugs, vulnerabilities or even malware.

3. Design and Implementation

NetReach preserves network availability by using Arm TrustZone to assign the network peripheral to the SW (Figure 1), thereby isolating it from all software in the NW and ensuring that no bug nor attacker can tamper with or disable that peripheral. We developed a proof-of-concept, based on OP-TEE OS [8], which provides firmware and software support for Arm TrustZone.

3.1. The Network Peripheral

Ensuring availability: NetReach protects the network peripheral by configuring the memory range(s) to which the peripheral or its associated bus-master is mapped to only accept read and write operations coming from the SW, as previously shown possible in [1], [2], [4], [6]. Additionally, NetReach assigns the interrupts of the peripheral to the SW and ensures that their priority is higher than any in the NW. We now have guaranteed (by the hardware) secure-only access and interrupt handling.

Transparent sharing is implemented using a split driver architecture, similar to the methods used in [1], [6]. The driver in the SW is responsible for mediating access to the hardware, while its NW counterpart simply forwards packets to the SW from the Linux kernel and back. These two drivers communicate via shared receive an transmit memory buffers and interrupts, enabling concurrent access and operation.

Interrupts assigned to each world enable asynchronous operation by signalling that receive and transmit operations have been performed on the buffers. By assigning interrupt priorities intelligently (low transmit priority for the NW), DOS attacks on the network in the SW are prevented.

Compared to [1], NetReach does not need a separate scheduler in the SW, rather relying on interrupt-based real-time operation. Additionally, NetReach minimizes its feature set in the SW to hardware access control and mediation (unlike [6]) to further reduce the TCB.

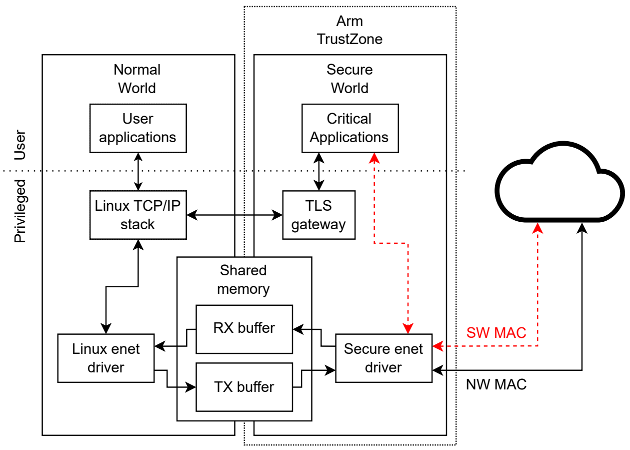

Figure 1. Current NetReach architecture.

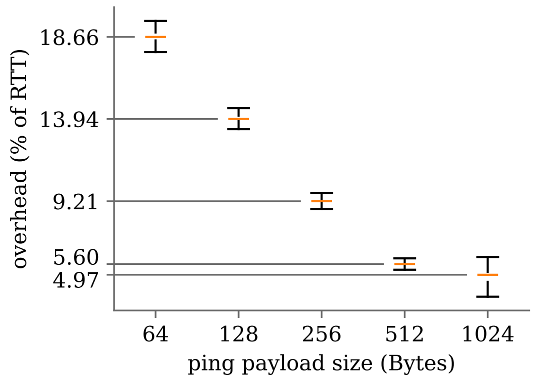

Figure 2. The median additional latency of moving the network peripheral to the Secure World is less than 20 % of the baseline RTT for small packets and to less than 5 % for larger packet sizes.

3.2. The Backup Network

When operating normally, NetReach uses features from OP-TEE OS to share the NW network stack with the SW, delivering flexible network support to critical applications. An added TLS encryption layer is provided by reusing cryptographic libraries already present in OP-TEE OS.

However, to **ensure reachability**, NetReach provides a backup network channel (red dotted line in Figure 1) which SW applications can dynamically switch to in case of attack and subsequent loss of the NW network connection; thus bypassing the fault in the network stack of the NW. The backup channel is assigned a separate MAC and IP address, cleanly separating its traffic from the NW. Additionally, it is possible for external entities to directly address these critical applications.

The backup network's capabilities are limited in order to minimize TCB code-size and thereby attack surface. However, commercial off-the-shelf Ethernet can support hardware implementations of common network- and transport-layer protocols[2]; NetReach thus supports simple UDP or TCP communication on the local network. Additionally, NetReach provides the cryptographic libraries necessary for symmetric encrypted communication, by reusing those already present in OP-TEE OS.

4. Evaluation

We implemented a proof of concept of NetReach on th BD-SL-i.MX6, a representative Armv7-A, Arm TrustZone-enabled processor [9]. An SPI based Ethernet Peripheral was used as the single network connection. The latency overhead of NetReach is presented below.

Network sharing overhead: We compared the Round Trip Time (RTT) with and without NetReach by using ICMP Echo Request/Reply (Ping/Pong) messages with varying payload sizes. The `ping` command in the NW provides this functionality. By measuring the time from the moment the SW is notified of a Ping in the transmit buffer, to the moment the SW notifies the NW of a Pong in the

[2] Like those made by WIZnet: https://wiznet.io/product/iEthernetChips

receive buffer, we have a baseline for the expected RTT. Due to the context switch, we observe a higher RTT in the NW: the median overhead ranges from **0.61 ms** for 64 byte payloads to **1.05 ms** for 1024 byte payloads. Compared to the total RTT, the overhead shrinks with increased payload size, from **18.66 %** to **4.97 %** (see Figure 2).

TCB size: NetReach increases TCB size by 418 Lines of Code (LoC), an increase of just **0.001 %** over the standard TCB Size of OP-TEE OS.

5. Conclusion

NetReach efficiently shares a network connection between mixed-criticality applications, ensuring guaranteed reachability to the embedded device even in cases where the NW stack is compromised. Our proof-of-concept implementation incurs a worst case overhead of 18.66 % for small packet sizes.

Acknowledgements

This research is partially funded by the Research Fund KU Leuven, the Cybersecurity Research Program Flanders, VLAIO (TRUSTI) and the European Union's Horizon 2020 research and innovation programme under grant agreement No 101020416 (ERATOSTHENES).

References

[1] T. Van Eyck, H. Trimech, S. Michiels, D. Hughes, M. Salehi, H. Janjuaa, and T.-L. Ta, "Mr-tee: Practical trusted execution of mixed-criticality code," in *Proceedings of the 24th International Middleware Conference: Industrial Track*, ser. Middleware '23. New York, NY, USA: Association for Computing Machinery, 2023, pp. 22–28. [Online]. Available: https://doi.org/10.1145/3626562.3626831

[2] J. Wang, A. Li, H. Li, C. Lu, and N. Zhang, "RT-TEE: Real-time system availability for cyber-physical systems using ARM TrustZone," in *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 2022, pp. 1573–1573.

[3] S. Wan, K. Sun, N. Zhang, and Y. Li, "Remotely controlling trustzone applications? a study on securely and resiliently receiving remote commands," in *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, ser. WiSec '21. ACM, Jun. 2021.

[4] L. Guo and F. X. Lin, "Minimum viable device drivers for arm trustzone," in *Proceedings of the Seventeenth European Conference on Computer Systems*, ser. EuroSys '22. ACM, Mar. 2022.

[5] W. Li, M. Ma, J. Han, Y. Xia, B. Zang, C.-K. Chu, and T. Li, "Building trusted path on untrusted device drivers for mobile devices," in *Proceedings of 5th Asia-Pacific Workshop on Systems*, ser. APSys'14. ACM, Jun. 2014.

[6] F. Schwarz, "Trustedgateway: Tee-assisted routing and firewall enforcement using arm trustzone," in *Proceedings of the 25th International Symposium on Research in Attacks, Intrusions and Defenses*, ser. RAID 2022. ACM, Oct. 2022.

[7] R. Ross, V. Pillitteri, G. Guissanie, R. Wagner, R. Graubart, and D. Bodeau, "Enhanced Security Requirements for Protecting Controlled Unclassified Information: A Supplement to NIST Special Publication 800-171," Tech. Rep., Feb. 2021. [Online]. Available: https://csrc.nist.gov/pubs/sp/800/172/final

[8] Linaro, "Open Portable Trusted Execution Environment," 2022. [Online]. Available: https://www.op-tee.org/

[9] Boundary Devices, "BD-SL-i.MX6 SBC." [Online]. Available: https://www.ezurio.com/single-board-computer/nxp-imx6/bd-sl-i-mx6