Ph.D. Project: Systems-on-Chip to Implement Zero-Trust Architectures

Abigail Butka
Department of Electrical and Computer Engineering
University of Florida
Gainesville, Florida
Email: butkaa@ufl.edu

Dr. Christophe Bobda
Department of Electrical and Computer Engineering
University of Florida
Gainesville, Florida
Email: cbobda@ece.ufl.edu

I. Problem and Motivation

With the prevalence of networked and embedded devices adopting System-on-Chip architectures there is a need for accompanying security architectures to protect these devices. Most current SoCs do not contain security built into the hardware, relying solely on external security. This makes it possible for any software application running on the SoC to read the sensitive data of all hardware components, perform Denial-of-Service attacks, and more. Given this, on-chip security is a hot topic for SoCs. One security architecture that is of international interest is the Zero-Trust Architecture.

In 2020 the National Institute of Standards and Technology (NIST) released their first Special Publication on Zero-Trust Architectures (ZTA) [1]. This publication details the core structure of a ZTA, as shown in Figure 1. While this architecture was primarily intended for networking applications, its concepts are widely applicable to related domains, such as embedded systems and SoCs.

A ZTA consists of an untrusted subject, a protected resource, and three components: Policy Enforcement Point (PEP), Policy Decision Point (PDP), and Policy Information Point. These components work together to determine whether a subject is trusted enough to access protected resources.

This publication by the NIST was followed in 2023 by a set of pre-prints focusing on implementing ZTA in commercial systems, for example, Volume B "Implementing a Zero Trust Architecture" [2]. To produce this publication, NIST worked with numerous industry partners who were interested in, or already produced, Zero-Trust products such as AWS Identity and Access Management, Cisco Firepower Threat Defense, and IBM's Security QRadar XDR. The wide range of companies involved showcases just how interested the community is in ZTA.

Motivated by the growing interest in ZTA principles across both research and industry, as well as the need for hardware security in SoCs, this work proposes a novel architecture that...

1) To our knowledge, is the first Zero-Trust Architecture implemented in an FPGA-based SoC.
2) Generically implements the requirements of NIST's Zero-Trust Architectures for application in CPS, IoT, etc.
3) Incorporates a Long-Short Term Memory (LSTM) deep learning algorithm that calculates the Trust-Level of users based on their transaction history.
4) Enforces Dynamic Mandatory Access Control in the form of Zero-Trust Policies for each component that state what Trust Level is needed to perform certain actions. For example, a Trust Level of 90% is needed to read the output of Component 1, but 50% for Component 2.

This prototype ZTA architecture is implemented as a SoC on a Xilinx Zybo Z-7010 FPGA SoC development board. It should be noted that our implementation does not include the PIP as it is intended to be a collection of third-party applications monitoring the system. These can be integrated into the decision-making of the PDP as needed.

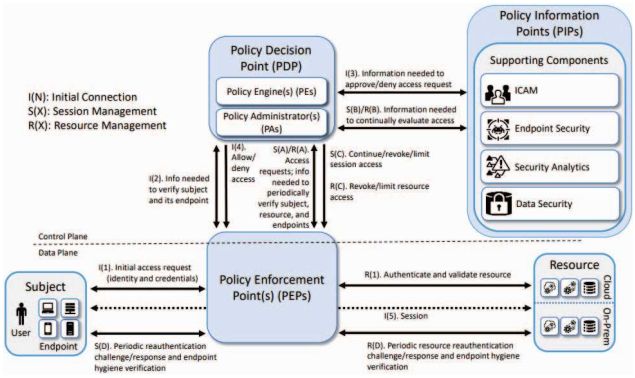

Fig. 1. NIST Zero Trust Architecture: Core Zero Trust Logical Components [2]

II. Related Works

In this section, we will discuss the state-of-the-art and related works on ZTA.

To begin, in [3], Yao et al. proposed Trust-Based Access Control, where the trust degree of the user is the deviation between the user's historical behavior and the current behavior. The higher the user's trust, the more authorizations they are given. Our work differs from theirs because while Yao et al. researched the methods and calculation of determining user

2576-2621/24/$31.00 ©2024 IEEE
DOI 10.1109/FCCM60383.2024.00058

trust, our work handles dynamically updating and implementing the AI model's decision.

Another related work comes from Zanasi et al. in [4]. The authors present a prototype ZTA architecture that is Software-Defined Network-based and updates component's security policies by distributing resource configuration files with certificates as proof of authorship. Our work differs from this work because our ZTA is implemented in the hardware and focuses on hardware-related challenges, rather than the setup and configuration of a SDN for ZTA.

The final related work is from Ferretti et al. in [5]. This work proposes a detailed exploration of Survivable ZTA for the cloud, where every component, including components that are typically considered trusted, can be compromised. This work differs from ours because it focuses on whether or not you can trust your own components, rather than the trust of a user.

III. Approach and Uniqueness

This section outlines the proposed architecture seen in Figure 2, which integrates the NIST's Policy Enforcement Point (PEP) and Policy Decision Point (PDP) into an SoC. While the implementation is different, this is the same architecture seen in Figure 1.

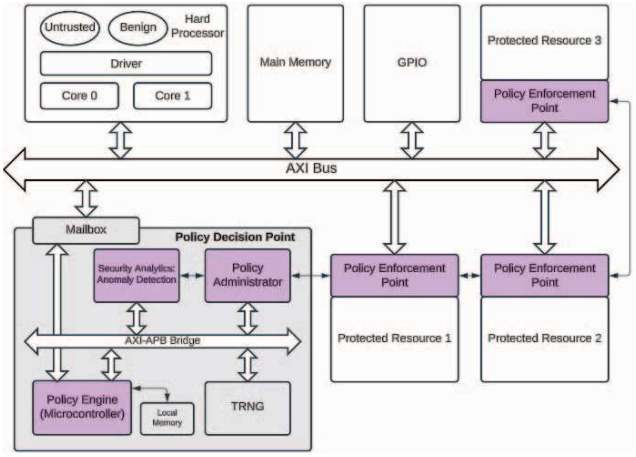

Fig. 2. Proposed Zero Trust Architecture

A. Policy Enforcement Point (PEP)

The proposed PEP is a dynamic firewall that ensures protected resources cannot be maliciously used by untrusted subjects. The PEP is placed between the protected resource and the untrusted subject to enforce a layer of separation, as seen in Figure 2. It then implements zero-trust policy checks on each access request. The implementation of the PEP consists of three components, the Access Vector Cache, the Policy Lookup Function, and the Enforcement Module.

1) Access Vector Cache (AVC): The AVC is a cache component that keeps the Policy Decision Point's most recently approved policy for its particular component. The AVC is regularly flushed and updated by the Policy Decision Point.

2) Policy Lookup Function (PLF): The PLF compares the policies in the component's AVC to the security policy provided by the untrusted subject. If the policy matches, the Enforcement Module (EM) is informed that the transaction is approved, and the transaction data is forwarded to the Policy Decision Point (PDP). If the policy does not match, the transaction is forwarded to the PDP for analysis and, once the PDP responds, the PLF forwards the decision to the EM.

3) Enforcement Module (EM): The EM enforces the decision sent by the PLF by either preventing or allowing the transaction to be forwarded to the component.

This implementation of the PEP is adequately suited to be used in Zero-Trust Architectures due to its capability to separate the resource from the network, dynamically update policies, and enforce the principle of least privilege, features which are critical to the implementation of security in ZTAs.

B. Policy Decision Point (PDP)

The proposed Policy Decision Point consists of two components, the Policy Engine and Policy Administrator.

1) Policy Engine: The Policy Engine runs incoming transactions through both a deep-learning LSTM and a user-defined set of zero-trust policies. The LSTM will determine what the Trust-Level of the user is over time given their prior usage of that component. The zero-trust policies are pre-defined and state whether a user can access that component, what functions or registers they have access to, and whether they can read or write.

2) Policy Administrator (PA): The PA handles dynamically updating the zero-trust policies of the Policy Enforcement Point based on the Policy Engine's decisions. The PA also handles enforcing the principle of least privilege within each PEP it controls, wiping their security policies after some amount of time.

IV. Expected Results and Contributions

The expected result of this work is the first working example of a Zero-Trust Architecture implemented on an FPGA-based SoC. Additionally, due to the versatility of both Systems-on-Chip and Zero-Trust Architectures, this work will, once finalized on an FPGA, be applied to the domains of Cyber-Physical Systems and Cloud-based Systems.

References

[1] V. Stafford, "Zero trust architecture," *NIST special publication*, vol. 800, p. 207, 2020.

[2] ——, "Implementing a zero trust architecture volume b: Approach, architecture, and safety characteristics (preliminary draft)," *NIST special publication*, p. 264, 2023.

[3] Q. Yao, Q. Wang, X. Zhang, and J. Fei, "Dynamic access control and authorization system based on zero-trust architecture," in *Proceedings of the 2020 1st International Conference on Control, Robotics and Intelligent System*, 2020, pp. 123–127.

[4] C. Zanasi, S. Russo, and M. Colajanni, "Flexible zero trust architecture for the cybersecurity of industrial iot infrastructures," *Ad Hoc Networks*, p. 103414, 2024.

[5] L. Ferretti, F. Magnanini, M. Andreolini, and M. Colajanni, "Survivable zero trust for cloud computing environments," *Computers & Security*, vol. 110, p. 102419, 2021.