

Platform Security Summit

Azure Sphere: A Secure IoT Platform

Jewell Seay

Principal Software Engineering Lead

Microsoft

SECURITY IS FOUNDATIONAL

It must be built in from the beginning.

Mirai Botnet attack

Everyday devices are used to launch an attack that takes down the internet for a day

100k devices

Exploited a well known weakness

No early detection, no remote update

Hackers attack casino

Attackers gain access to casino database through fish tank

Entry point was a connected thermometer

Once in, other vulnerabilities were exploited

Gained access to high-roller database

<https://www.washingtonpost.com/news/innovations/wp/2017/07/21/how-a-fish-tank-helped-hack-a-casino/>

OS Bug exposes 200M Devices

RTOS TCP Flaw








Core OS code flaw

Impacts medical, elevators, modems, SCADA

Most devices have no method to be updated

The 7 properties of highly secured devices

<https://aka.ms/7properties>

- | | | |
|---|---|---|
|  | Hardware Root of Trust | Unforgeable cryptographic keys generated and protected by hardware. Physical countermeasures resist side-channel attacks. |
|  | Small Trusted Computing Base | Private keys stored in a hardware-protected vault, inaccessible to software. Division of software into self-protecting layers. |
|  | Defense in Depth | Multiple mitigations applied against each threat. Countermeasures mitigate the consequences of a successful attack on any one vector. |
|  | Dynamic Compartments | Hardware-enforced barriers between software components prevent a breach in one from propagating to others. |
|  | Certificate-Based Authentication | Signed certificate, proven by unforgeable cryptographic key, proves the device identity and authenticity. |
|  | Error Reporting | Renewal brings the device forward to a secure state and revokes compromised assets for known vulnerabilities or security breaches. |
|  | Renewable Security | A software failure, such as a buffer overrun induced by an attacker probing security, is reported to cloud-based failure analysis system. |

Meeting these seven properties is difficult and costly

Design and build
a holistic solution



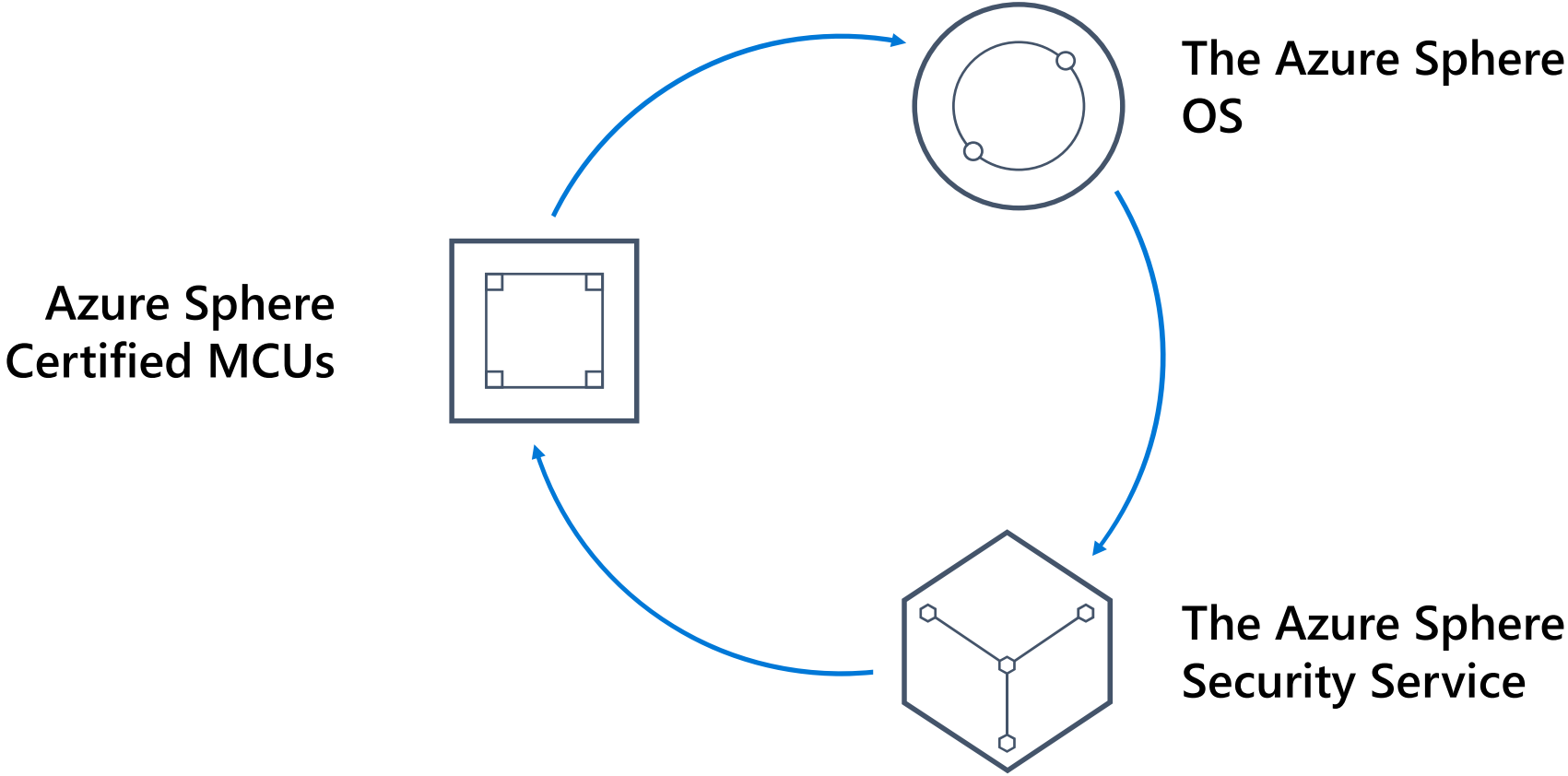
Recognize and mitigate
emerging threats



Distribute and apply
updates on a global scale



Azure Sphere is an end-to-end solution for MCU powered devices



The Azure Sphere OS Architecture

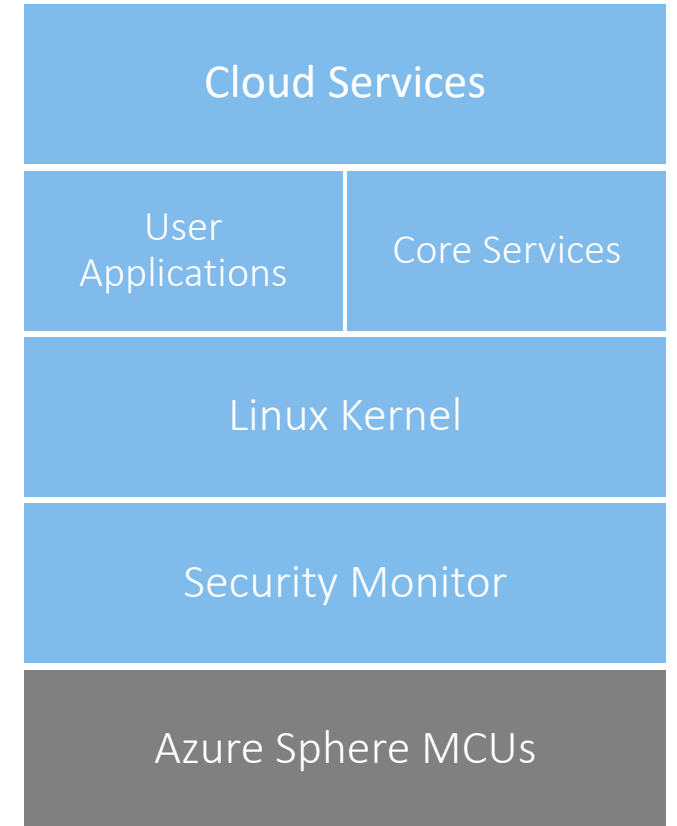
Cloud Services for immediate full device software updates

Core Services to support updates, cloud, and network communications

Linux Kernel to leverage the open source community's security reviews and small footprint capabilities while leveraging existing platform code

Security Monitor for monitoring the security of the software platform

Azure Sphere MCUs for hardware-based security design



Azure Sphere Hardware Layout

Hardware Firewall for full chip to chip communication control

Pluton our core of the chip security design

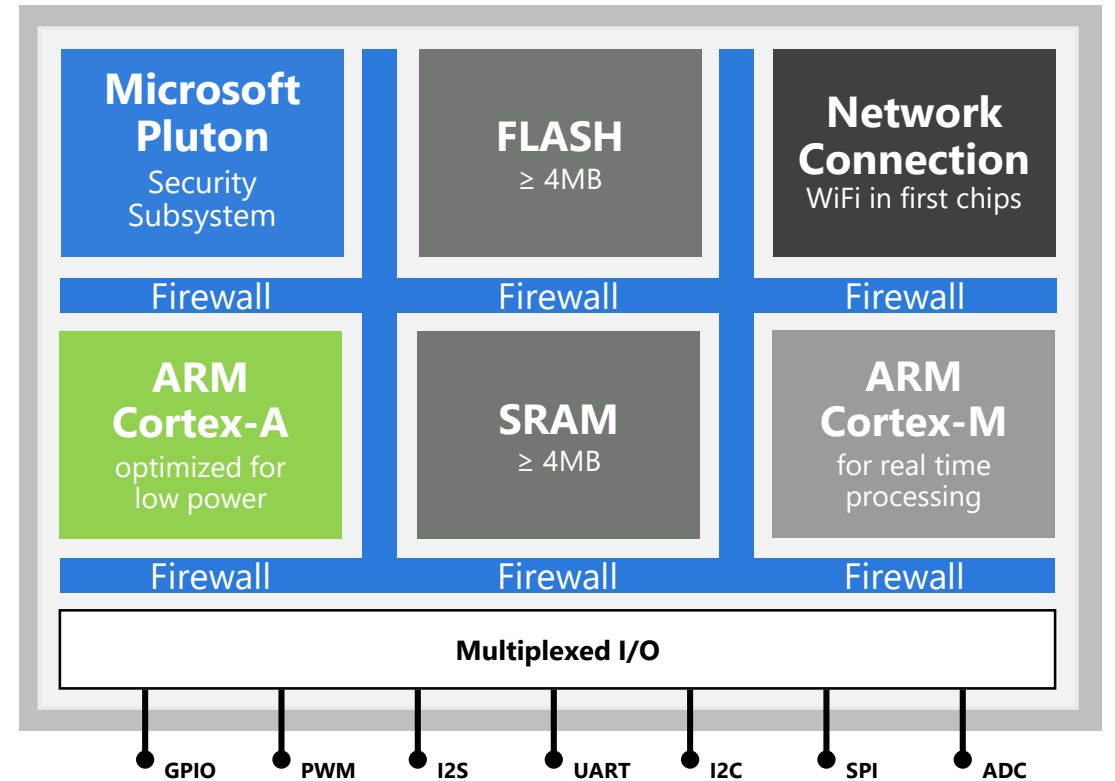
Onboard Wifi built-in for network connectivity

ARM Cortex-A for running the Security Monitor, Linux Kernel, and user applications

ARM Cortex-M for running a RTOS providing immediate IO access when demanded

FLASH for storage of all images and data on the system

SRAM to provide memory for executing applications

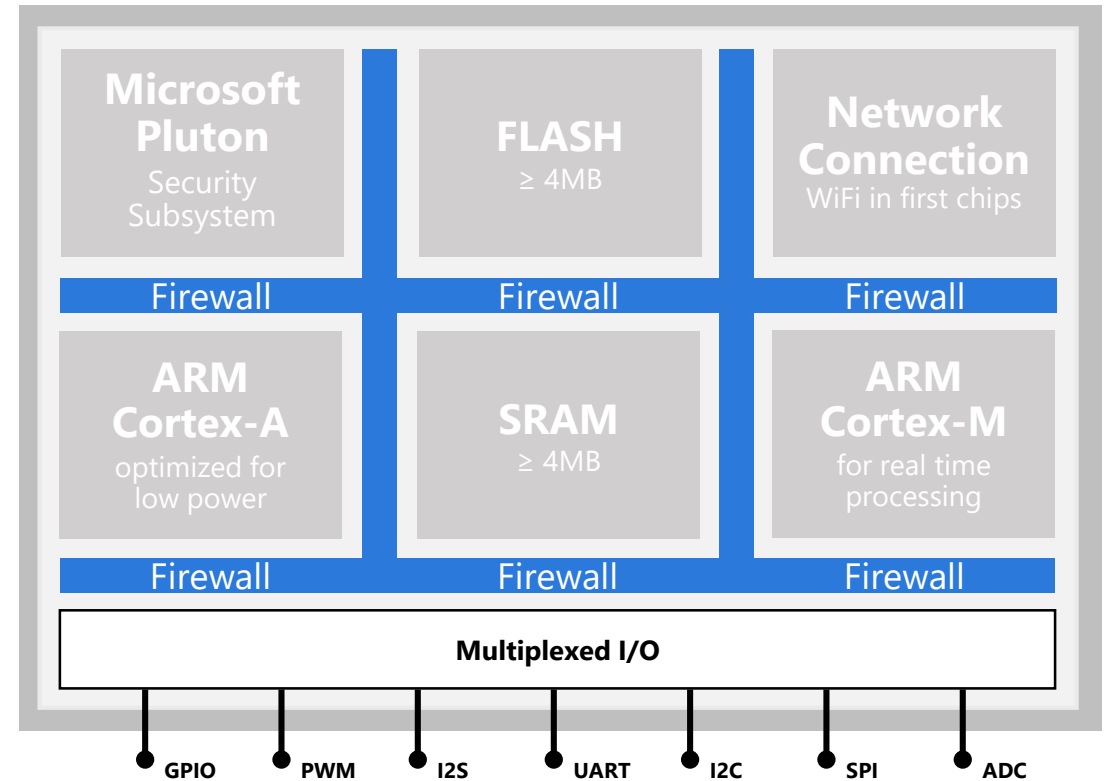


Azure Sphere Hardware Firewall

Configurable firewalls support authenticated access to various peripherals and memory regions allowing protection of the memory map

Limited control to only security monitor for changing the configuration, locking down the regions on boot for RAM and key flash resources

Lock bits used to lock a configuration until SoC reset preventing an escalation of privilege from reconfiguring the firewalls



Defense in Depth: Pluton & Security Monitor

Dedicated M4 CPU for Hardware Root of Trust

Public/Private keys hardware generated when chip is set to a secure state during manufacturing

Private keys burned into e-fuses, only accessible by the hardware encryption module

Public Key securely transmitted to AS3

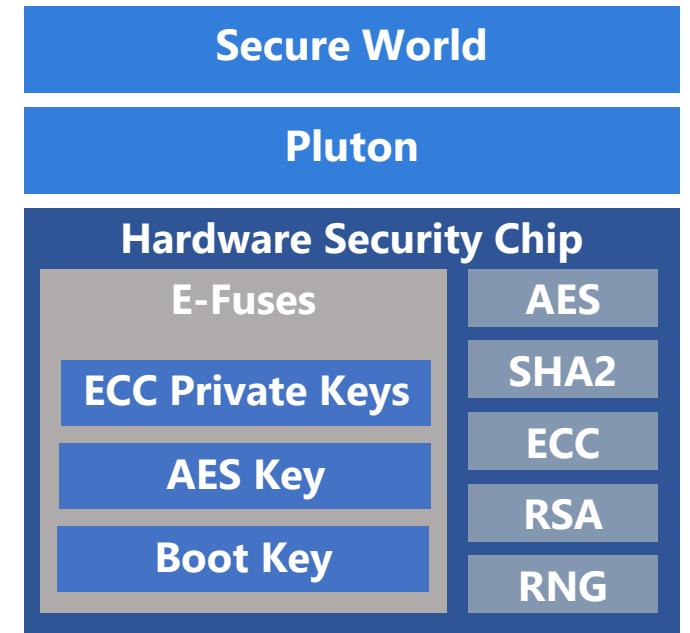
Encryption hardware only accessible by the Pluton subsystem

Flash writes can only be accomplished by Pluton

Pluton validates and boots Security Monitor

Security Monitor validates and boots the Linux Kernel

Application Signatures are verified by SM and Pluton before Linux Kernel loads an application



Linux Kernel

Kernel.org is source of kernel code

4.9.x release branch with movement to newer LTS branches as stability is proven

Upstream releases are merged monthly

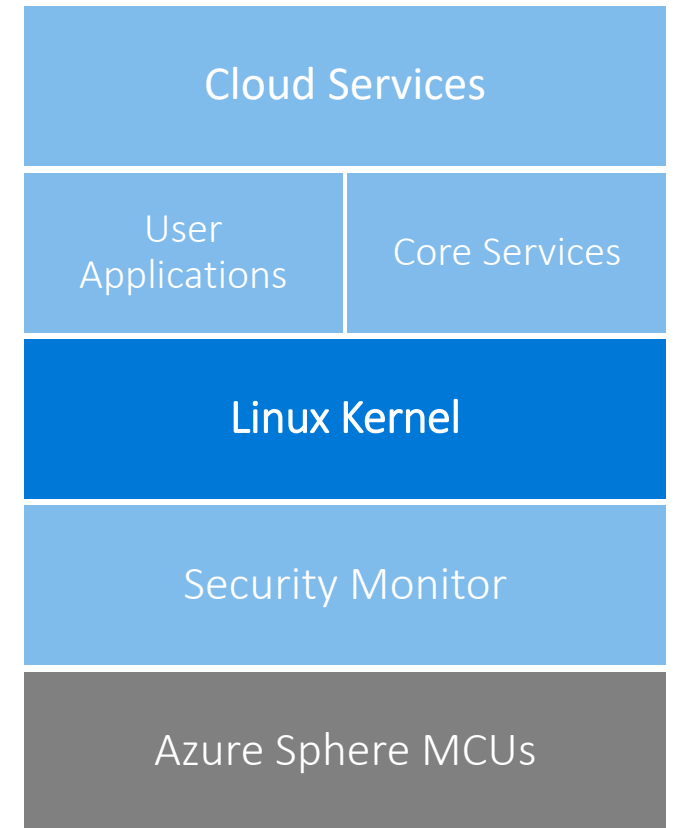
No module support to keep kernel binary size small and limit attack surface

No runtime code generation due to kernel size and memory impact

No sudo type functionality as no application needs to change user IDs

Custom LSM for process credentials

Disable page execution flag if the memory page was ever marked writable in the past



Linux Userland

No shell or console as there is no purpose

Leverage OSS Libraries like WolfSSL, cURL, and musl

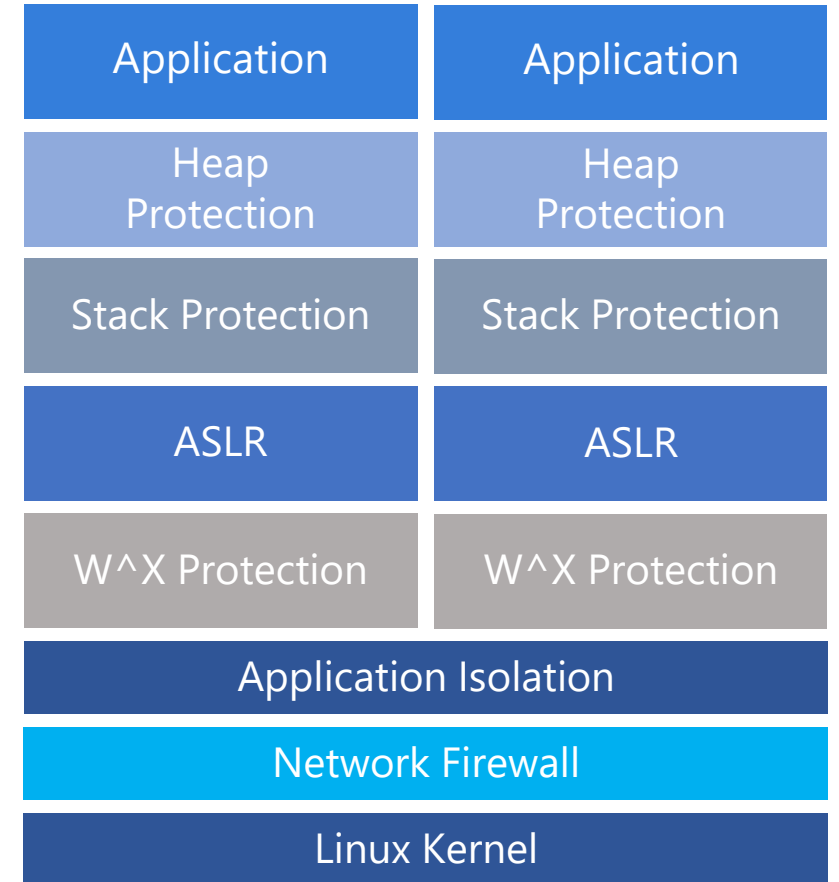
Linux user isolation forcing each process to run as a separate user id including all core services

Custom init process that does not run as root, only root processes are kernel threads

Latest GCC compiler version used with default enablement of ASLR, stack protection, and non-executable stack

Network Firewall rules are application specific allowing control of IP and allowed domains to be compiled into the application manifest as part of the application package

Default deny of inbound and outbound network traffic



Cloud Services

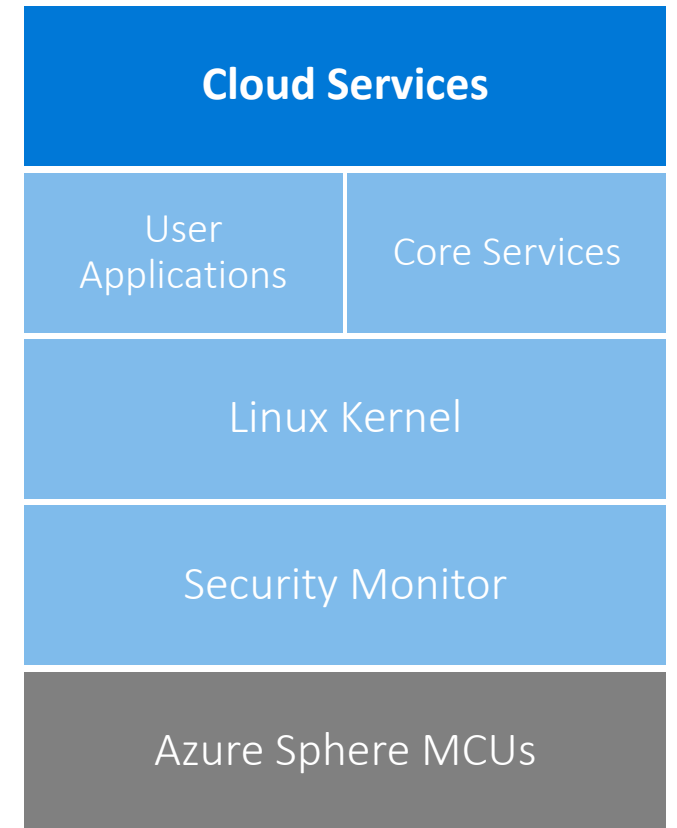
Attestation measured at the beginning of boot and modified by hashes of core components

Non-writable register stores Attestation value, can only be updated by encryption hardware and requires a full SoC reset to start over.

AS3 Verification of the Attestation, combined with a nonce and signed with the device private key to confirm device identity and software. Device told to update if verification fails or old software detected.

AS3 TLS Authentication provided by a short lived (24h) certificate after attestation is confirmed allowing AS3 service access

Daily Check-in allows for rapid deployment of critical security concerns along with monthly full system updates



Build System

Yocto build framework

Automated tests ran on each build including static code analysis and fuzzing

Functionality tests done with emulation and on actual hardware

All tests must pass to allow for Pull Request

Nightly CVE report for used open source components, 48 CVE patches so far in 2019 and 6 component version upgrades outside of Yocto release schedule

Linux SACK Panic pushed to the release pipeline within 2 days of kernel patches becoming public

Security flags validated on all Linux components, an extended version of Checksec

C++17 comprises the majority of internal code, complicating string, buffer, and array-based attacks. Very few objects inherit limiting object confusion and C++ provides automatic memory and object handling limiting heap attacks

