2024 WHITE PAPER

ARCA TRUSTED OS

For Raspberry Pi 4B - Overview

CYSEC SA - EPFL Innovation Park - CH 1015 Lausanne
www.cysec.com

CYSEC

A Trusted Execution Environment for containers running on Raspberry Pi 4B at the edge

Raspberry Pi 4B is a small, versatile and cheap electronic board that is more and more used in industrial environments. It is mainly used for prototyping, however a growing number of companies use them for pre-production or small production series. When used in production, Raspberry Pi 4B boards need to be combined with appropriate software components to implement a trustworthy and secure device. These properties are especially important when these devices are remote edge devices deployed in the field like, for example, gateways collecting data in public areas.
CYSEC has developed ARCA Trusted OS for Raspberry Pi 4B with the objective to bring significant levels of trustworthiness and security to a Raspberry Pi-based device.

1 - Overview on ARCA Trusted OS for Raspberry Pi

ARCA Trusted OS for Raspberry Pi is a hardened Linux micro-distribution designed to host and run containerized applications. This OS embeds standard container toolings (Docker, k8s) and it can be installed on Raspberry Pi 4 boards.

In a nutshell, ARCA Trusted OS for Raspberry Pi 4B is a Trusted Execution Environment (TEE) to operate containers at the edge. CYSEC approach is to enforce security at the OS level to protect the containerized applications and their data. ARCA Trusted OS can cope with some of the security and operational challenges that end-users are facing when operating containerized applications in distributed architectures at the edge.

The design of this Trusted Execution Environment has been defined with four main security objectives in mind :

- To prevent the exploitation of potential vulnerabilities of ARCA Trusted OS in order to compromise the application and its data, its attack surface has been reduced to the minimum needed to host and run containers.
- To ensure the trustworthiness of the infrastructure hosting and running containers in the edge, ARCA Trusted OS relies on a trusted computing base (TCB).
- To protect data stored in SD cards of remote devices and data communications within edge networks, ARCA Trusted OS embeds by default data protection mechanisms.
- To keep ARCA Trusted OS for Raspberry Pi 4B up-to-date and secure in time, it comes with a secure update tool for applying CYSEC security maintenance patches.

Furthermore, ARCA Trusted OS for Raspberry Pi 4B benefits from CYSEC security maintenance process that leverages on:

The regular ARCA Trusted OS for Raspberry Pi 4B release lifecycle;
An extensive vulnerability management process;
A robust and secure patch/release process.

2- Security features

This section is structured around the four security objectives described in the Overview section. For each security goal, the design of the security features that have been implemented is described in detail.

2.1 Attack surface minimization

Arca Trusted OS is a purpose-built operating system for hosting containers. Therefore, the hardening strategy can be pushed further than in the case of a traditional general-purpose OS.

The two main ways to achieve this are :

- Reducing the attack surface to the minimum by including only the kernel modules and software packages required to run the operating system and containerized applications.
- Enforcing a more restrictive default configuration to reduce the possibilities of misconfigurations impacting the OS security.

Those two principles are applied at two levels: the Linux kernel, the user space and system configuration.

a) Kernel configuration

The kernel hardening of Arca Trusted OS relies on a minimalistic approach, where only the kernel modules that are required to deploy containers and run all the components of ARCA Trusted OS are added. Furthermore, none of the kernel modules are loaded at runtime, which ensures that security-related modules cannot be deactivated at runtime. In addition, several available security features for ARM kernels have been activated, such as,e.g., the deactivation of the access to the kernel configuration from the user space, some mitigations against memory based buffer overflows, the randomization of memory cache and page allocation.

b) User space hardening

Similarly to the kernel, only the user space tools that are needed to deploy and access containerized applications have been included. From a network point of view, the OpenSSH daemon is the main way to interact with the system. Therefore, its configuration is particularly curated by enforcing best practices on SSH session credentials, lifetime and forwarding. Finally, strict default configurations are applied to file permissions and user login policies.

2.2 Trusted computing base for container platforms

This section describes what makes ARCA Trusted OS a Trusted Execution Environment at runtime.

a) Immutable system executables and software libraries

Concerning security, the main advantage of a read-only file system is to restrict the lateral movements an attacker can do. Particularly, this prevents the attackers from installing kernel modules or software packages at runtime, guaranteeing the system integrity at runtime. While it's definitely not a way to prevent all types of attacks, it still helps mitigate the most common attack paths.
Another advantage of a read-only file system is its capacity to make the operating system more reliable. Indeed, a read-only file system lowers the probability that one or several critical files get corrupted (e.g. due to a misconfiguration or an operational error). Therefore, the system is way less likely to enter a state where it is not able to boot anymore. In the case of a host OS for containers, one needs to allow end users to be able to deploy and operate their containers without the restrictions of a read-only file system. Therefore, the user file systems where the containers and data are stored need to keep read/write.

Furthermore, ARCA Trusted OS for Raspberry Pi 4B allows the customization of some configurations (e.g. the interface network configurations) hosted in read-only file systems using the `overlayfs` implementation. The `overlayfs` implementation keeps the customized configurations after a reboot or an update of the operating system. In order to meet the needs described above, CYSEC makes the choice to make all system executables and software libraries read-only in the boot and root file systems, whereas it allows modification of some configurations in the root file system by its end-users and offers a dedicated space where containers can be deployed and write data.

The user file system partition of ARCA Trusted OS for Raspberry Pi 4B, called the data partition, is formatted with BTRFS. Its main purpose is to store user data. By default, the BTRFS partition is divided into three subvolumes that are mounted as read/write overlays (overlayfs). BTRFS has been selected for its high resilience to data corruption (Copy on Write and checksums).This approach gives the ability to users to store data in their home directories and edit/add system configuration to the /etc path as usual. The /var path is also writable because several applications, such as Docker, require being able to `write` data in it.

A `side` benefit of having read-only file systems is that those file systems are made squashfs partitions in ARCA Trusted OS, compressing these file systems to minimize their sizes in the SD card.

b) Secured boot chain

A secure boot chain is used to ensure the authenticity and integrity of the operating system running on the machine. The end goal is to avoid an attacker replacing the operating system by a vulnerable or backdoored copy in order to tamper with the system or steal confidential information.
Usually, a secure boot chain is achieved by verifying the signature of each step of the boot process to ensure that only binaries that have been signed with a trusted key can be launched. In the case of the ARCA Trusted OS for Raspberry Pi 4B, the key pair used for the authenticity verification is proprietary to CYSEC but unique to each end user. The public part of this key pair is stored in the One-Time-Programmable (OTP) memory of the Raspberry Pi board. ARCA secure boot chain is enforced by default. Another particularity of ARCA Trusted OS secure boot is that it covers all read-only partitions. This allows the authentication and the verification of the integrity of the boot and the root file systems at each boot making the booted operating system trustworthy for its users.

ARCA Trusted OS for Raspberry Pi 4B secure boot chain starts with the power up of the Raspberry Pi SoC. This SoC verifies the integrity of the secure boot public key stored in the EEPROM with its hash value fused on the OTP memory. Then, the SoC uses this secure boot key to verify the authenticity and integrity of the ARCA Trusted OS bootloader before launching it. This boot loader is responsible for verifying the authenticity and integrity of the FIT image containing the kernel and the initramfs before launching them. And finally, the initramfs is in charge of verifying the integrity of the root file system before launching it and the user file systems.

2.3 Data protection

This section presents the security mechanisms that CYSEC puts in place to protect the data stored in or transmitted from/to ARCA Trusted OS for Raspberry Pi 4B

a) Full disk encryption

In order to protect the user data at rest, the user file system is encrypted. This prevents an attacker who steals the board from being able to retrieve sensitive data stored on the SD card. This protection is applied at the OS level to provide data protection by default to the containers without the risk of misconfigurations from the end users.
To achieve this encryption, Linux Unified Key Setup version 2 (LUKS2) is used. LUKS2 encrypts and decrypts data on the fly when going from the SD card to the RAM. This means that the data are only decrypted on demand, and only in clear when stored in the RAM or in the processing unit. For this reason, even if an attacker can gain physical access to the SD card of a running system, it won't be able to retrieve the data in clear.

As the device will be deployed in the field, it is important that the encryption key used by the FDE is protected from physical attacks and that LUKS-managed partitions can be unlocked automatically and autonomously each time the device boots. One solution matching those criteria is the use of a Trusted Platform Module chip (TPM) to protect the key and to release it only after a successful secure boot.
As TPM can sometimes be hard to source or to add to an existing hardware design, an alternative solution can be exploited. On the Raspberry Pi 4B boards, some OTP memory space is dedicated to the storage of secrets and can be used to securely store the LUKS key.

b) Network security via Wireguard

By default, Arca Trusted OS for Raspberry Pi 4B is shipped with all the user space tools to use the in-kernel WireGuard implementation. While it does not add any layer of security in itself, it incentivizes the users to build a secure network architecture using WireGuard in use cases involving distributed edge devices. This secure network architecture enforces authenticated communications and protects the confidentiality and integrity of the data in transit.

2.4 Security maintenance

This section presents the update mechanism chosen by CYSEC to allow end-users to simply keep ARCA Trusted OS for Raspberry Pi 4B up to date. Furthermore, it describes the security mechanism used to prevent updating to an unapproved OS version.
ARCA Trusted OS includes an OTA (Over The Air) update mechanism based on the open-source piece of software swupdate. With OTA updates, manufacturers and developers can deliver software patches, security fixes, feature enhancements, or even entirely new firmware versions to embedded devices wirelessly, eliminating the need for manual intervention and physical connectivity. ARCA Trusted OS OTA update mechanism can easily be automated and put at scale by using management tools such as Ansible.
An update for ARCA Trusted OS for Raspberry Pi 4B comprises the boot file system and the root file system only. The user file systems, including containers, are not updated when ARCA Trusted OS is. In the implementation of ARCA Trusted OS update mechanism, two main objectives were prioritized: the provision of a redundant operating system fallback mechanism and the support for authenticated updates.

a) Redundancy implementation through an A/B scheme

In the implementation of redundancy of ARCA Trusted OS for Raspberry Pi 4B, the boot file system and the root file system are duplicated in the SD card layout. ARCA Trusted OS for Raspberry Pi 4B boot logic provides a mechanism for automatic fallback to the alternate partition if the currently selected one fails to boot successfully. This ensures system availability and resilience even in the event of a boot failure.

b) Authentication

In ARCA Trusted OS, the authentication of software updates is implemented by signing the sw-description file rather than individually signing every component of the .swu package. This approach offers efficiency and ensures the integrity of the update process. The sw-description file contains the boot file system and the root file system. By signing the sw-description file, any alteration to these file systems would result in the digests no longer matching with the signed file. The pair of RSA keys used for the authentication of the software update is proprietary to CYSEC. During the update process, the signature verification is the initial step performed. If the signature cannot be verified using the public key (stored in the root file system) or if the images do not match the digests specified in the sw-description file, the update is aborted to prevent any potential tampering or unauthorized modifications.

3 - Description of CYSEC maintenance processes and tools

This section is structured around the three security maintenance processes described in the Overview section. For each security maintenance process, the design of the process steps that have been implemented is described in detail

3.1 ARCA Trusted OS release cycle

In the realm of operating systems, the notion of a "Secure Operating System" necessitates the implementation of frequent and regular software package updates. This imperative characteristic arises due to the ongoing discovery of security vulnerabilities in various software components. Any operating system, regardless of its initial "secure by design" claims, becomes susceptible to these vulnerabilities over time if it fails to adapt and address them in a timely manner.

It is worth mentioning that ARCA Trusted OS for Raspberry Pi 4B may have multiple variants to cater to different requirements. The mainline version of the OS is not tailored to any specific customer but serves as a general version. Additionally, there are customer-specific variants that are precisely customized to meet the unique needs of individual clients. Therefore, the infrastructure supporting the release cycle needs to support the existence of multiple builds, signatures, and updates.

a) The release process : Leveraging the Yocto Project

In the release process of ARCA Trusted OS for Raspberry Pi 4B, the CYSEC development team relies on the Yocto Project framework for development, building, and maintenance purposes. This framework offers several advantages, one of which is the ease of upgrading packages. The CYSEC team has decided to leverage the concept of "layer" within the Yocto project.

A Yocto layer is a collection of packages. By updating specific layers (typically around 5 layers), the entire operating system can be updated. This streamlined approach simplifies the upgrade process.

Once the operating system has been upgraded with the latest packages, the CYSEC team proceeds to prepare the release. A release consists of an OS disk image and an update package (covering only the boot file system and the root file system). To ensure the integrity and authenticity of these components, every element is signed.

b) Delivery and repository Management

The regular release cycle of ARCA Trusted OS for Raspberry Pi 4B is based on a frequency of at least one release every three months, maintaining a regular release cycle. However, in case of the identification of a vulnerability having a critical impact on ARCA Trusted OS hot security fixes are rapidly released after the availability of a patch.

The delivery of ARCA Trusted OS updates takes place on a CYSEC dedicated update repository, where the OS disk image and update package are uploaded as soon as they have been signed.

3.2 Vulnerabilities management

CYSEC vulnerability management process relies on some particularities of the Yocto project, and it is structured in 4 steps:

- Monitoring: collection of the CVEs publicly known.
- Identification: listing of the CVEs impacting ARCA Trusted OS for Raspberry Pi 4B packages
- Assessment: qualification of the impact of the identified CVEs to prioritize their mitigation
- Mitigation: strategic and operational approaches for fixing the identified CVEs

a) CVE monitoring with Yocto project tooling

The Yocto project maintains its own infrastructure and a comprehensive database that houses publicly available Common Vulnerabilities and Exposures (CVE). This database is regularly updated to ensure the latest information regarding security vulnerabilities.
Within the CYSEC team, a detailed report is generated during the project build process. This report provides a list of the packages used in the ARCA Trusted OS for Raspberry Pi 4B project, including their respective versions and any potential CVEs that may impact these packages.

CYSEC maintenance team relies on this CVE report to monitor the state of each package in terms of vulnerabilities. This CVE report is generated on a daily basis and is used as input to the CVE identification step.

b) CVE identification

To effectively identify the CVEs that impact ARCA Trusted OS for Raspberry Pi 4B, the CYSEC team has developed a Vulnerability Aggregator Framework. This framework aims at listing all the CVEs impacting one or more ARCA Trusted OS for Raspberry Pi 4B packages at a given time.
This framework relies on the CVE report as its input to generate a database of the identified CVEs. The Vulnerability Aggregator consumes a fresh CVE report every night to keep the CVE database up-to-date with the latest information regarding vulnerabilities.

c) CVE assessment

In terms of prioritization and urgency, the CYSEC team will evaluate each vulnerability based on its severity. If a CVE is deemed critical or poses a significant risk, an urgent release can be initiated. This prioritization mechanism allows the team to focus their efforts on addressing and mitigating high-risk vulnerabilities promptly.

At the first step of assessment, the severity of the CVEs is based on the score given by the NIST on its database. If the value of this score is above 7, CYSEC performs further analysis to evaluate with precision the impact of this CVE on ARCA Trusted OS as described in table 1.

SEVERITY	SCORE	ACTIONS
Low	0.1 - 3.9	· No communication to the customers when the CVE is assessed · Wait for the upstream patch and integrate it on the next release
Medium	4.0 - 6.9	· No communication to the customers when the CVE is assessed · Wait for the upstream patch and integrate it on the next release
High	7.0 - 8.9	· No communication to the customers · Refinement of the impact assessment on ARCA Trusted OS for Raspberry Pi 4B. · If the vulnerabilitly is applicable to ARCA Trusted OS, a fix will be present for the next release, either by the upstream branch or by a manual patch
Critical	9.0 - 10.0	· Refinement of the impact assessment on ARCA Trusted OS for Raspberry Pi 4B to check if the vulnerability is effectively present and applicable to the OS (false positive sanity check) · If NO : wait for the upstream patch. No need to do an extra release · If YES : Integrate the upstream patch if any of manually ASAP for a special release

Table 1 : Actions taken to assess and mitigate CVEs depending on their severity

d) CVE Fixing / Mitigation

When addressing vulnerabilities in a package, there are two primary methods available. The first approach involves upgrading the package itself. In this case, the original maintainers of the package have identified and resolved the vulnerability, providing an upgrade for users. The second method involves applying a patch to the vulnerable version of the package. These patches are typically sourced from the Yocto layers maintainers, although there are instances where the CYSEC team may develop their own patches.

Upgrading the Yocto layers to their latest versions is CYSEC's main strategy for resolving the majority of vulnerabilities. The timing decision of the generation of a new release including updated Yocto layers or the integration of specific patches is defined in table 1.

As part of the release process, the team will provide a summary of the vulnerabilities that have been addressed and fixed. This information will likely be included in the Release Note section of the ARCA Trusted OS for Raspberry Pi 4B documentation.

3.3 Signature process

In order to enable the secure boot features described earlier, the ability to sign each part of the system and ensure the confidentiality of the keys is paramount. Therefore, the signature process is completely under Cysec’s control and follows strict security requirements.

When designing the signature process, CYSEC wanted to ensure that :

- Only authorized CYSEC employees can perform a signature
- Firmware that is signed has always been reviewed by multiple peers
- The presence of more than at least 3 CYSEC employees is required to sign an update.

a) Design and type of keys

Before describing the release signature process, it is important to list and characterize the keys used within this process. Since each stage of the boot has different requirements, either linked to the hardware or the software framework, three different keys are needed to sign each part of the system.

A hash of the FSBL (First Stage BootLoader) keys is fused into the OTP chip of the board. This hash is used to verify the integrity of the FSBL keys stored in the EEPROM memory. It means that once the OTP is fused and closed, only FSBL that have been signed with the same key can boot on the device.

The FIT image key is hardware-agnostic and is the same for each customer and board type. This key is stored in the ARCA Trusted OS bootloader.

The types of keys are shown in table 2 :

KEY USAGE	TYPE OF KEY	SPECIFICITY	KEY STORAGE
RPI 4B FSBL	RSA 2048 bit	Client specific	· Secret key : derived from CYSEC shares · Public key : stored in EEPROM · Public key hash : stored in OTP
FIT IMAGE AUTHENTICATION	RSA or ECDSA	Shared by all the clients	· Secret key : derived from CYSEC shares · Public key : stored in the bootloader
SWUpdate	RSA	Shared by all the clients	· Secret key : derived from CYSEC shares · Public key : stored in root file system

Table 2: Table listing the key usages, the key types and their specificity

b) Signature process description

In order to protect the signature keys presented above, they are stored encrypted on a dedicated signature computer that is kept offline at all times and stored in a secure location.

All the keys are encrypted using AES and, in order to avoid one person having access to this key, it is split into multiple shares thanks to the Shamir secret sharing (3 shares out of 5).

The shares are then dispatched between different departments of the company.

The signature process follows the workflow detailed below:

All the system artifacts are build on the Yocto build server from the release branch
The artifacts are uploaded to the artifacts' repository
All the artifacts are transferred to the signing laptop (offline machine containing the signing keys)
Each shareholder inputs its share of the encryption key to decrypt the signature keys.
The FSBL keys for each client are derived from the master secure boot key.
The artifacts are signed on the signing laptop.
The signed artifacts are used to generate the final image(s) and the update image

c) Signature tooling

The tool used for all signature and key generation operation is proprietary and developed fully by CYSEC to perform the signature of ARCA Trusted for Raspberry Pi 4B.

- End of the document -

ABOUT CYSEC

Founded in 2018 by experienced cybersecurity experts, CYSEC is a European cybersecurity company, counting more than 35 employees, with offices in France, Switzerland and Italy.
We help companies to securely deploy their applications with highly sensitive data in industries such as critical infrastructure, edge applications, healthcare, defence and space.
Indeed, CYSEC is pioneering end-to-end, European and off-the-shelf cybersecurity products for newspace missions.

CYSEC SA
EPFL Innovation Park, Building D
CH- 1015 Lausanne, Switzerland

info@cysec.com

www.cysec.com

www.linkedin.com/company/cysecsystems