Future-Proof uCPE Security with Arm TrustZone and Telco Systems NFVTime

WHITE PAPER – FEB 2019

Introduction

NFV opens a world of new business opportunities for Communication Service Providers (CSPs), however there are several new security challenges that are inherent to the dNFV/uCPE architecture. This document will outline some of these concerns and illustrate how a multi-level, combined hardware/software security approach can be used successfully mitigate these new threats. This document will focus on secure Universal CPE (uCPE) using implementation methods based on Arm TrustZone capabilities in partnership with Telco Systems' NFVTime uCPE software suite.

Telco System's NFVTime

NFVTime is an open and neutral plug-and-play uCPE suite delivering a complete service environment for the smooth and rapid launch of NFV services, along with a service lifecycle management and operational tools. NFVTime enables CSPs to immediately begin deploying scalable, customized VNF services, offering on-demand managed business services, On-Net and OTT/Off-Net.

uCPE Security Concerns and Vulnerabilities

uCPE is a remote NFV node, deployed in a non-secure or controlled environment utilizing remote automated installation processes such as zero touch provisioning, which introduce additional set of vulnerabilities that must be evaluated and addressed. The security concerns that we will review in this document will cover:

- Multi-Vendor Software Stack
- Open, loosely coupled software environment
- Virtualization: Hosting a Blackbox OS
- Remote installation and provisioning

Multi-Vendor Software Stack

The legacy CPE appliance was based on custom-made hardware and a monolithic software image, both typically supplied by the same CPE vendor. However, with the uCPE concept there exists a collaborative multi-vendor environment, with hardware and software often provided by many different vendors. This multi-

vendor nature is related both to the fact that a uCPE is expected to run any 3rd-party VNFs and to the fact that that the NFVi itself may be built from a mix of proprietary and open-source components (e.g. Linux, KVM, OVS, OpenStack). Even though each of these components individually may be well-tested for security vulnerabilities, the combination of components may create new and unexpected vulnerabilities.

Open, loosely coupled software environment

To ease the interoperability and complexity, the combination of hardware and software from multiple vendors requires loose coupling via open (well-known and documented) interfaces. For example, OpenStack uses the well-documented Libvirt API to control the KVM Hypervisor. OpenStack itself provides a well-known Northbound API, while its internal APIs (between the controller and the Agent parts) may be easily obtained due to the open-source nature of the project. The fact that such mission-critical APIs, that are becoming a de-facto standard for pure-play NFVi implementations, are open and well-documented, creates multiple new threat vectors and exploitation options for malicious actors. There's no longer a need for complex reverse-engineering in order to understand how an API works and try to find its vulnerabilities. In short, we may be making things a lot easier for the hackers.

Virtualization: Hosting a Blackbox OS

At the heart of the uCPE architecture resides a virtualization layer, that allows not only 3rd party code, but a whole OS with a set of applications on top that runs on the host device. A VNF is essentially a software Blackbox and any uCPE is expected to be able to run any such Blackbox, which is a potential vulnerability.

Firstly, a potential attack on the VNF may compromise the security of the uCPE (and not just of a single isolated hardware appliance as in the past). Secondly, due to the open nature of the uCPE, a hacker gaining access to the device, or even to an external API such as OpenStack API, may be able to inject and run a malicious VNF image without being detected.

Remote installation and provisioning

Remote Zero-Touch Provisioning (ZTP), along with automated remote updates and upgrades, has become a standard requirement for any uCPE deployment. These capabilities are critical in order to deliver on the NFV promise of major OPEX reduction and operations simplification. However, such automated process eliminates any authentication and verification steps performed by authorized personnel of the CSP. This automation may introduce security vulnerabilities and bi-directional attack vectors:

1. From device into network — hacking into the network by using a device controlled by a hacker
2. From the network to the device — hacking into the device by installing a hacker-controlled 'fake' uCPE provisioning server that impersonates the real one

uCPE Attack Vectors

The following actions may be performed by malicious actors on a uCPE device to either damage the network's proper function, make illegitimate use of network resources, or tap into and potentially exfiltrate sensitive CSP data including end-user information. These actions may become easier to implement on a uCPE (vs. legacy physical CPE) due to the factors described in the previous section.

Security Threat	Possible ways to Implement
Hijacking network resources	• Hijack / steal uCPE device • Provision unauthorized uCPE device
DOS — denying of service to end-user	• Destroy / modify SFC • Destroy VNFs
Critical VNF bypass (e.g. vFW bypass)	• Modify SFC
Steal sensitive information and secrets	• Install "Tap VNF" • Modify SFC • Install malicious NFVi component(s)
Inject malicious traffic – viruses or DDOS attacks into the network	• Install malicious VNF • Modify SFC • Install malicious NFVi component(s)
Attacks on VNFs: steal VNF licenses, security keys or other sensitive data	• Install malicious VNF • Gain full access / hack into uCPE

NFVTime-OS Multi-Layer Security Model

NFVTime-OS implements a multi-layer security model, comprising of various complementing protection mechanisms. Each mechanism by itself is designed to provide protection against specific attacks, collectively providing a comprehensive uCPE security solution.

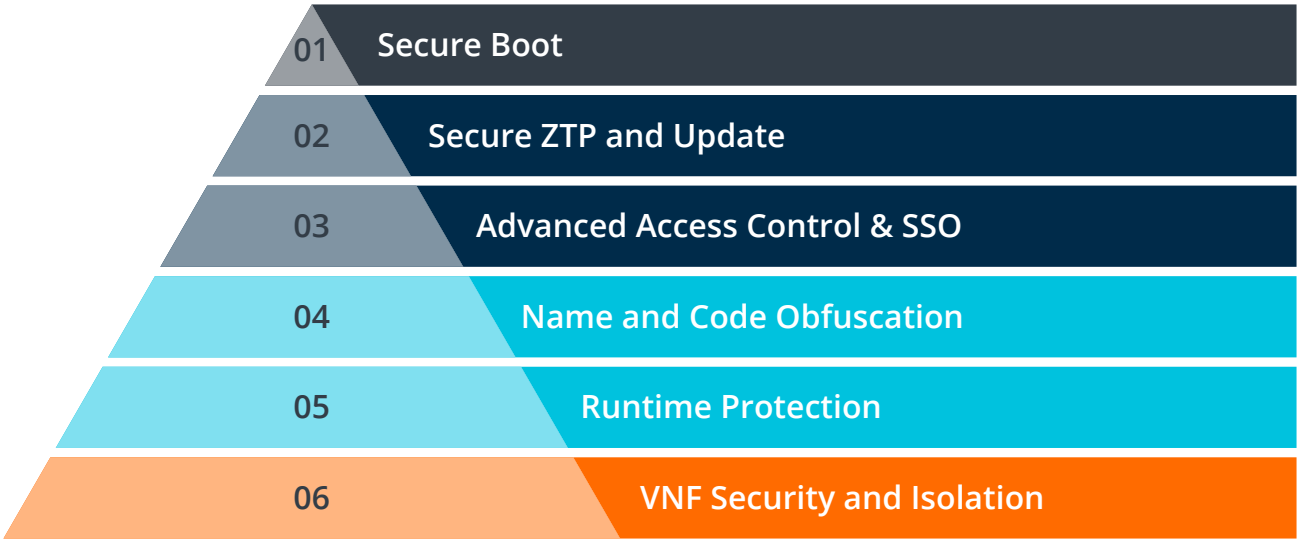

Figure 1. NFVTime-OS Multi-Layer Security Model

Description of NFVTime-OS Security Layers

Extended Verified Secure Boot

Verified secure boot is a well-known and widely implemented software protection mechanism, which verifies that the software image that boots (in this case NFVTime-OS) is indeed the verified software image provided by uCPE vendor (Telco Systems), and that is has not been tampered with. The secure boot mechanism is essential to uCPE due to the fact many of the open source software components (e.g. Linux Kernel, KVM or even OpenStack) could be replaced with hacked versions.

Verified secure boot creates a Chain-of-Trust, preferably rooted in hardware, where each component in the chain cryptographically verifies the next signed component before transferring control to it. The verification is done based on the encrypted signature of the original image. If any of the components in the chain are modified, secure boot will detect this change and will terminate the boot process.

The existing Verified Boot implementations usually stop the Chain-of-Trust at the Linux Kernel, which is the last component being verified. However, in the uCPE case, this still leaves plenty of room for attacks, since higher-level components such as OpenStack are critical for proper operation of uCPE and the provisioned network services. For this reason, Telco Systems plans to extend the Chain-of-Trust all the way up to the VNF image level, and include all the critical components of NFVTime-OS, as well as its verified configuration, in this chain.

- ## Secure ZTP and Update

  Securing the ZTP process and any remote upgrade/update process is critical to prevent attacks where either the uCPE or the Provisioning Server (i.e. NFVTime uCPE Manager) is replaced by an unauthorized component or by a "man-in-the middle" attack. It is essential to implement advanced two-way authentication and authorization mechanisms based on unique certificates, as well as cryptographic hashing and validation functions, to securely verify the contents of every package downloaded to the uCPE from the Provisioning Server (i.e. every NFVTime-OS image, every configuration file and every VNF image).

- ## Advanced Access Control and SSO

  NFVTime-OS is built from several field-proven open-source components, many of which (such as Linux or OpenStack) allow user access. In order to better secure the uCPE against potential unauthorized user logins, NFVTime-OS implements a unified Single Sign-On (SSO) mechanism that manages all user rights for all software components in one place, as supports RADIUS and TACACS+ remote authentication mechanisms.

- ## Runtime Protection

  Verified boot (see Layer-1) verifies the critical code and data at boot time. Once the uCPE boots successfully with all its services and VNFs, the verified boot process ends. However, this does not protect against any potential attacks that change the critical code or data during run-time. For example, an attacker may gain access to the uCPE and replace OpenStack code in memory, to enable a backdoor to run unauthorized VNFs. This is a very advanced type of attack which requires advanced protection. Runtime protection seeks to solve this potential vulnerability by performing periodic validation checks of all critical software code and data – within uCPE memory and storage devices, during run time.

- ## Name and Code Obfuscation

  Code obfuscation may prevent attackers from reverse-engineering the code. However, since part of any NFVi-OS is open-source, the original code can always be obtained anyway, given the hacker knows which open-source components are used and what to look for. Hence name obfuscation is even more important for uCPE, when the names of the files and libraries are scrambled and encrypted, it's much harder to detect which components (either open source or proprietary) are utilized within NFVTime-OS. Consequently, it is more difficult to replace a single component without immediate detection, or reverse-engineer the OS.

- **VNF Security and Isolation**

  VNF Security refers to a future capability to better protect the VNFs by allowing them to utilize the underlying Security Services of NFVTime-OS, including hardware-based security isolation mechanisms. This will both protect VNFs from hackers and from each other. Such capabilities are currently under research and may require future hardware security technologies, such as Arm Secure Partitions.

Utilizing Arm TrustZone™ to Enhance uCPE Security

Arm TrustZone™ is a hardware-based security isolation mechanism that is present in modern Arm Cortex-A architecture SoCs. It creates an isolated Secure World which can be used to provide confidentiality and integrity to the system. Trusted applications or sensitive data can be installed and stored in the Secure World for maximum protection and isolation, as they become totally inaccessible to the code running in the Normal World (the rest of the OS & application code) via CPU hardware protection mechanisms.

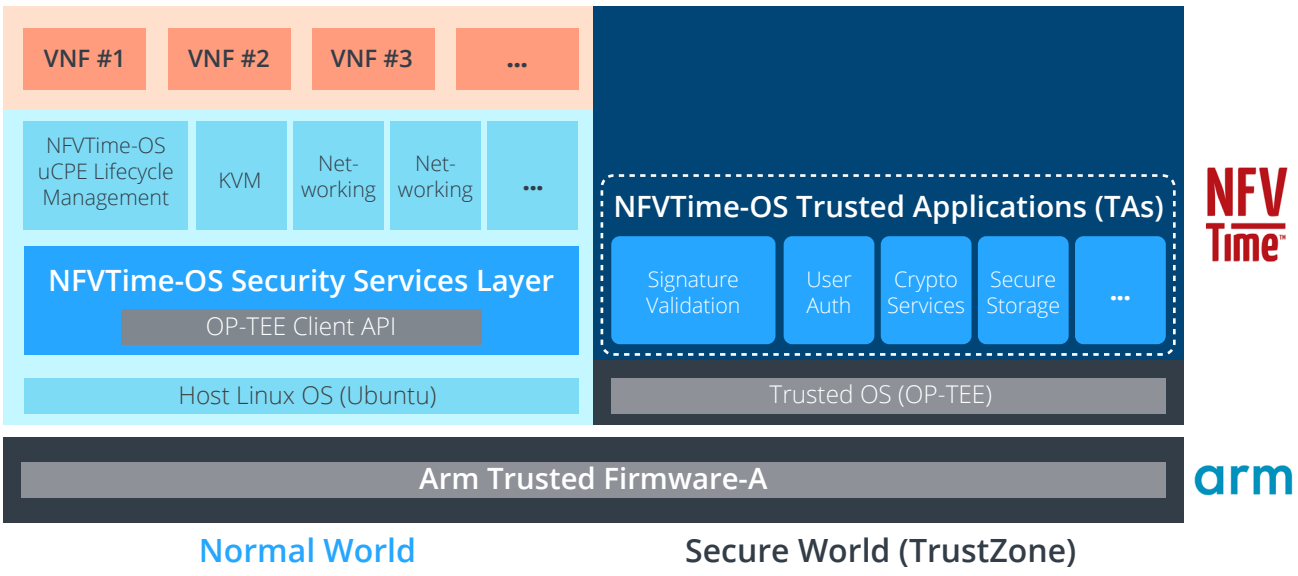

Figure 2. NFVTime for Arm Security Architecture, utilizing Arm TrustZone Technology

Applications installed in Secure World can only be invoked from the Normal World via special protected APIs, which emit a Secure Monitor Call (SMC) instruction that is handled by the Secure Monitor executing in TrustZone. TrustZone hardware makes sure that no code running in the Normal World gets direct access to code or data in the Secure World. The Secure Monitor enforces which secure functions can be invoked, and if allowed, the caller will only get back the result from the trusted function call once it finishes its execution. Moreover, code running in the Secure World is considered part of the system firmware and is maintained in flash

along with other firmware components. The firmware upgrade process requires authorization, firmware image validation, and a system reset. Any unauthorized changes will be automatically detected by verified boot process.

This means that Arm TrustZone can be efficiently utilized to isolate sensitive code and data, and provide the security services essential for securing the uCPE. With other architectures, critical OS security services (see below) would fully run in software within the same address space (from CPU perspective) as the rest of the NFVi-OS. When running on Arm they'll be fully isolated by hardware in the Secure World. This makes it much more challenging to modify or remove the code that implements these services or retrieve any secrets or sensitive information stored in the Secure World by the OS.

Telco Systems has developed a timeline to implement the following critical low-level security services within the Secure World when running on Arm based uCPEs:

- **Hardware Root of Trust** – Any secure boot process must start with an initial RoT (Root of Trust). Arm Trusted Firmware-A (TF-A) provides BL1, boot room (the initial boot loader) which is immutable code that runs in Secure World and verifies the first mutable firmware component. This verification process is repeated for all firmware components, thus establishing a Chain of Trust.
- **Extended Verified Secure Boot** – The verified boot process begins with the initial RoT (as described above) and continues with verifying and loading the other Arm Trusted Firmware-A boot components. Standard UEFI Secure Boot continues the boot process and includes verifying and loading the OS kernel. NFVTime-OS will further extend this process to include all critical OS components beyond the Linux Kernel, and up to and including the VNF images themselves. The additional components will boot in the Normal World (as usual), but whenever the next component in the chain has to be verified, the OS will call a Trusted Application running in the Secure World (via a Secure Monitor Call) to perform the validation and return the result.
- **Code & Image Verification** – will be used both by Extended Verified Boot as described above, and by Secure ZTP / Upgrade functions of NFVTime-OS. The downloaded image signature verification may be based on device identity (Certificate / MAC / Serial Number) as an additional security measure, and to make sure the device gets the image that was indeed intended for it. This verification function can be also used to verify OS configuration (full or partial) – provisioned or stored on the device. For example, this will help ensure no changes have been unknowingly made to the SFC (Service Function Chain) configuration and prevent VNF bypass.

- **Access Control** – authentication/authorization of either human users or remote servers (e.g. NFVTime Controller or uCPE Manager), in case local RBAC (Role-Based Access Control) is used. This trusted application will both encrypt the whole user data base, and run the authentication and authorization code in the Secure World.
- **Security State & Attestation** – if NFVTime-OS detects a compromise or verification mismatch, it may ask this designated internal security service, which is fully protected by Arm TrustZone, to change the security state of the uCPE device. The current security state can be fully trusted and may be later queried and retrieved by a trusted authority for attestation purposes.
- **Secure Storage** services to encrypt and store sensitive data and secrets (e.g. Certificates) based on a pre-generated private key, which itself is stored in the Secure World. Since both the key and the encryption/decryption functions will run within the Secure World boundary, the data itself may be stored anywhere on the uCPE storage device, or in RAM, and will still be safe.
- **Generic Cryptography Services (Symmetric and Asymmetric)** for encryption and decryption of certificates, signatures, keys, passwords and other sensitive data as required by NFVTime-OS and its various components and applications. Cryptography services may be used in particular for Code and Name obfuscation.
- **Key Generation and RNG (Random Number Generation)** services.

Summary

Migration from legacy CPE appliances to uCPE based network deployments, while promising many technological and business advantages, also introduces many new security risks. These risks are related to the inherent uCPE architecture principles and design goals and require innovative security measures as part of the uCPE or NFVi implementation itself. Telco Systems delivers enhanced multi-layer uCPE security solution as part of its NFVTime Suite and NFVTime-OS. NFVTime for Arm takes a major leap forward in enhancing uCPE security by efficiently utilizing Arm TrustZone technology, which isolates and protects critical OS security services and assets in a hardware protected partition (the Secure World). Combining an Arm-based uCPE device with NFVTime for Arm software suite delivers an advanced, secure uCPE solution, well protected against both existing and future attacks.