Real-time Kernel Protection (RKP)

The security of the kernel is essential to the security of the whole system. An attack that compromises the kernel has the ability to arbitrarily access system sensitive data, hide malicious activities, escalate the privilege of malicious user processes, change the system behavior or simply take control of the system. As mentioned previously, Trusted Boot measurements can be used to determine what kernel was loaded and run when the device was started. However, this protection does not guarantee the integrity of the kernel after the system runs and starts to interact with potential attackers. Clever attackers can sometimes exploit an already booted and running kernel. In such cases, it is important to continuously monitor the kernel during the system *runtime* in order to detect and prevent modifications to the kernel code or critical data structures.

Intuitively, the kernel protection mechanism cannot itself exist completely in the kernel, or it could be circumvented by an attacker. Therefore, Samsung KNOX introduces Real-time Kernel Protection (RKP), a unique solution that provides the required protection using a security monitor located within an isolated execution environment. Depending on the device model, this isolated execution environment is either the Secure World of ARM TrustZone or a thin hypervisor that is protected by the hardware virtualization extensions. RKP's Trusted Computing Base (TCB) is part of this isolated environment and thus is secure from attacks that may potentially compromise the kernel.

Running in an isolated execution environment may cripple the ability of the security protection mechanisms to closely monitor events that happen inside the target kernel. To solve this problem, RKP uses special techniques to take full control over the Normal World memory management and intercept critical events and inspect their impact on security before allowing them to be executed. Hence, RKP complements TIMA-PKM's periodic kernel integrity checking, which has limited effectiveness against attacks that can take place and properly hide their traces between periodic checks.

RKP achieves three important security features:

- First, RKP completely prevents running unauthorized privileged code (i.e., code that has the kernel privilege) on the system, which is accomplished by preventing modification of the kernel code, injection of unauthorized code into the kernel, or execution of the user space code in the privileged mode.

- Second, RKP prevents kernel data from being directly accessed by user processes. This includes preventing double mapping of physical memory that contains critical kernel data into user space virtual memory. This is an important step to prevent kernel exploits that map kernel data regions into malicious processes where they could be modified by an attacker.

- Third, RKP monitors some critical kernel data structures to verify that they are not exploited by attacks. In particular, RKP protects the data that defines the credentials assigned to running user processes to prevent attackers from escalating this credential by modifying this data.