Abstract :

Secure elements store and manipulate assets in a secure way. The most attractive assets are the cryptographic keys stored into the memory that can be used to provide secure services to a system. For this reason, secure elements are prone to attacks. But retrieving assets inside such a highly secure device is a challenging task. This paper presents the process we used to gain access to the assets in the particular case of Java Card secure element. In a Java Card, the assets are stored securely, i.e., respecting confidentiality and integrity attributes. Only the native layers can manipulate these sensitive objects. Thus, the Java interpreter, the API and the run time act as a firewall between the assets and the Java applications that one can load into the device. Finding a vulnerability into this piece of software is of a prime importance. Finding a vulnerability into a software is often not enough to develop a complete exploit. Here, we demonstrate at the end that a Java Card applet can call the hidden native functions used to decipher the secure container that encapsulates a key. Some previous attacks have shown the ability to get access to the application code area. But the Java Card intermediate byte code detected in the dumps has shown several differences with regard to the specification, which prevents the reverse engineering of the applicative code. Thus, to avoid the execution of shell code by a hostile applet, a part of the byte code stored into the card is unknown. The transformation is done on-the-fly during the upload of an application. We present in this article a new approach for reversing the unknown instruction set of the intermediate byte code which in turn has led to reverse engineering of the Java classes of the attacked card. We discovered during the reverse that some method calls have an unusual signature. Without having access to the native code, we have inferred the semantics of the called methods and their calling convention. These methods have access to the assets of the card without being restricted by security mechanisms like the firewall. We exploit this knowledge to set up a new attack that provides a full access to the cryptographic material and allows to reset the state of the card to the initial configuration. We demonstrate the ability to call these methods at the Java level in an application to retrieve sensitive assets whatever the protections are. Then, we suggest several possibilities to mitigate these attacks.