Hardware Security Modules not immune to hacking

Reference: Memo [190612-1] Date: 12/06/2019 - Version: 1.0
Keywords: HSM, cryptography, PKCS#11, cloud, digital services
Sources: publicly available information

Key Points

- Security researchers released a paper revealing how they managed to hack a Hardware Security Module (HSM).
- HSM-s are used to generate, manipulate and store sensitive cryptographic secrets (SIM cards, credit cards, secure boot hardware, disk and database encryption, PKI…).
- HSM-s are also used by cloud service providers, such as Google or Amazon, allowing clients to centrally create, manage and use their cryptographic secrets.

Summary

Researchers from a French technology company, specialising in blockchain and cryptocurrency ecosystems, published and presented a paper detailing how they managed to dump the whole content of a Hardware Security Module (HSM), manufactured by an undisclosed company. In order to achieve this, they proceeded as follows:

1. They started by using legitimate software development kit (SDK) access to their test HSM to upload a firmware module that would give them a shell inside the HSM. Note: A SDK is typically a set of software development tools that allows for the creation of applications for a certain software package, software framework, hardware platform, or computer system.
   Note that this SDK access was used to discover the vulnerabilities, but is not necessary to exploit them.
2. They then used the shell to run a fuzzer on the internal implementation of PKCS#11 commands to find reliable, exploitable buffer overflows. Note: PKCS#11 refers to public-key cryptography standards or to a programming interface used to create and manipulate cryptographic tokens.
3. They verified that they could exploit these buffer overflows from outside the HSM, i.e. by just calling the PKCS#11 driver from the host machine.
4. They then wrote a payload that would override access control and, via another issue in the HSM, allow them to upload arbitrary (unsigned) firmware. It is important to note that this backdoor is persistent – a subsequent update will not fix it.
5. They then wrote a module that would dump all the HSM secrets, and uploaded it to the HSM.

The attack leverages several vulnerabilities, including code execution, in the module and its PKCS#11 implementation to gain administration privilege. Using this method, the HSM's dump can be decrypted offline and all its contents can be revealed.

On June 10, the Gemalto Company confirmed that the affected HSM is the Safenet Protect Server PSI-E2/PSE2. The vulnerabilities have now been patched.

Comments

The HSM vulnerable to the attack is a FIPS 140-2 Level 3 validated module. This standard, written by the US NIST, lists a number of requirements that the HSM and its documentation must comply with. The attack described in the research paper illustrates that compliance and validation of a HSM according to a well-known security standard are not sufficient conditions to guarantee that it is actually secure.

It is plausible that well-resourced / state-sponsored entities could have carried out similar work and discovered this attack.

An interesting aspect of the attack is that the firmware update backdoor is persistent. There can plausibly be live HSM-s deployed in critical infrastructure now containing similar backdoors.

HSMs can take various shapes like smartcards, PCI extension cards or components embedded in various equipment (smartphone, computer, network appliance, server, etc.) and are being used every time sensitive cryptographic secrets must be generated and manipulated (SIM cards, credit cards, secure boot devices, disk and database encryption, PKI…).

These components are also available in cloud-based versions, hosted by companies like Google or Amazon, allowing clients to centrally create, manage and use their cryptographic secrets using APIs. This type of product may be used to store secrets related to cloud-based encrypted hosting but its use poses several concerns, namely loss of control over such a critical part of the security infrastructure (HSM's geographical location and ownership) and the lack of knowledge related to the implementation (is the HSM shared or not, administrator access, brand/model…).