www.utimaco.com

utimaco®

Utimaco IS GmbH • Germanusstraße 4 • 52080 Aachen

To whom it may concern

26.08.2015

Company Statement concerning key extraction vulnerability CVE-2015-5464

Utimaco has been made aware of vulnerability CVE-2015-5464, which is summarized as follows: "*The Gemalto SafeNet Luna HSM allows remote authenticated users to bypass intended key-export restrictions by leveraging (1) crypto-user or (2) crypto-officer access to an HSM partition*."
Random Oracle's article On Safenet HSM key-extraction vulnerability CVE-2015-5464 (part I) and Safenet HSM key-extraction vulnerability (part II) provide insights into details of this vulnerability, and describe possible mitigations and workarounds.

Background
As Random Oracle's article states *"[…] PKCS#11 specification. This is a de facto standard designed to promote interoperability between cryptographic hardware by providing a consistent software interface. […] PKCS#11 is a very complex standard with dozens of APIs and wide-range of cryptographic operations, called "mechanisms" for everything from encryption to random number generation. Safenet vulnerability involves the key derivation mechanisms."*

Conclusion
The vulnerability CVE-2015-5464 is entirely based on functions and mechanisms specified in the PKCS#11 standard, in particular the C_Derive function with mechanism CKM_EXTRACT_KEY_FROM_KEY. Hence, all standard-compliant PKCS#11 implementations supporting these mechanisms are affected. Whether a given application is actually subject to this vulnerability depends on the specific environment and setting of key usage flags.

Measures
Utimaco strongly encourages all users of our Hardware Security Modules to take the following measures.

- Do not rely on default settings for key usage attributes. Instead, explicitly disable non-intended usage by setting the respective attribute to CK_FALSE. The attribute CKA_DERIVE must be set to CK_FALSE to thwart this specific vulnerability.
- In general, limit the allowed usage of a key to only those mechanism(s) the key shall actually be used for. Set key attributes such as CKA_ENCRYPT, CKA_DERIVE, or CKA_SIGN to CK_FALSE if a key is not foreseen to be used for encryption, key derivation, or signing.

- If the application uses a key derivation function as part of its intended operation, it is of course impossible to completely disable key derivation by setting CKA_DERIVE to CK_FALSE. In this case, we advise paring down the key derivation functions that can be invoked to the required minimum via CKA_ALLOWED_MECHANISM.
- Disallow modification of attributes by setting CKA_MODIFIABLE to CK_FALSE. Ensure that all attributes have been set as intended prior to setting CKA_MODIFIABLE to CK_FALSE; once CKA_MODIFIABLE has been set to CK_FALSE, attributes cannot be changed anymore.

Key usage attributes should be defined explicitly during key generation or import, and the attribute CKA_MODIFIABLE should be set to CK_FALSE at this very moment. If you are not sure about the attribute settings of your keys, we recommend to run Utimaco's PKCS#11 Administration Tool P11CAT and proceed as follows:

1. Select the PKCS#11 slot you want to inspect, and login as user.
2. Open "Object Management".
3. Double-click on a key object to display its attributes
4. If necessary, alter key usage attributes as described above.
    - Attributes will be changed instantly when selecting a new value.
    - Modification is only possible while CKA_MODIFIABLE is set to CK_TRUE.
5. Set CKA_MODIFIABLE to CK_FALSE. From now on, attributes cannot be changed anymore.

In addition, we recommend to apply state-of-the-art active security mechanisms including, but not limited to, virus scanners and intrusion detection systems. Always keep user credentials confidential. Whenever possible, avoid storing PKCS#11 passwords in configuration files.

With best regards
Utimaco IS GmbH

Dieter Bong Product Manager HSM	Thorsten Grötker Head of R&D HSM