CipherShed - Bug #107

Open Crypto Audit Project TrueCrypt CS-TC-1 - CryptAcquireContext may silently fail in unusual scenarios

04/05/2015 02:35 AM - Jason Pyeron

Status:	New	**Start date:**	04/05/2015
Priority:	Immediate	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:		**Spent time:**	0.00 hour

Description

1. CryptAcquireContext may silently fail in unusual scenarios
Class: Cryptography Severity: High Difficulty: Undetermined
FINDING ID: CS-TC-1
TARGETS: CryptAcquireContext calls in Random.c
DESCRIPTION: The parameters passed to CryptAcquireContext may cause it to fail in certain obscure situations. TrueCrypt calls CryptAcquireContext in the following manner:

```
if (! CryptAcquireContext (& hCryptProv , NULL , NULL , PROV_RSA_FULL , 0)
&& !CryptAcquireContext (& hCryptProv , NULL , NULL , PROV_RSA_FULL , CRYPT_NEWKEYSET))
CryptoAPIAvailable = FALSE;
else
CryptoAPIAvailable = TRUE;
```

Listing 1: Calls to CryptAcquireContext

Testing on Windows XP indicates that if this is the first time a user has issued a call with the NULL container (parameter 2), the first call to CryptAcquireContext will fail, while the second, initializing a new KeySet, will succeed. A later version of Windows tested appears to succeed on the first call, but this was not thoroughly tested.
While disturbing, this issue should not cause failure on common Windows XP uses. However, this is not the correct method of calling CryptAcquireContext and it may cause failure on uncommon Windows configurations (spanning XP through Windows 8.1).
CryptAcquireContext acquires a context to a user's key container to store keys in; however, TrueCrypt does not use it for that purpose – rather it uses it exclusively for generating random numbers. In certain circumstances (such as Mandatory Profiles [https://groups.google.com/forum/#!searchin/microsoft.public.platformsdk.security/CryptAcquireContext/microsoft.public.platformsdk.security/4dJc5eVeywA/qAaUy2xWNy8J]) a key container cannot be initialized and the call will fail.
Even though TrueCrypt does not need to store keys, it will be unable to generate random numbers.
To address the situation where an application does not need to persist keys, the CRYPT_VERIFYCONTEXT flag is available and should be used. When present, CryptAcquireContext will not attempt to access a user's key container, and therefore will not fail if it could not do so.
This problem is exacerbated by the fact that the application does not fail if it cannot acquire a handle to a Cryptographic Service Provider – it will simply continue without strong randomness, and use other poor values of randomness such as Process ID and various pointers. More detail about the RNG in the absence of calls to CryptGenRandom is covered in Appendix A on page 17.
EXPLOIT SCENARIO: A user creates a TrueCrypt Volume on a company-managed machine. Because of the Group Policy Settings in place at the organization, TrueCrypt is unable to open a handle to a Cryptographic Service Provider, and falls back to insecure sources of randomness, potentially enabling brute-force attacks on the master key.
Recommendation: Pass the CRYPT_VERIFYCONTEXT flag to CryptAcquireContext rather than attempting to create a new keyset. If CryptAcquireContext or CryptGenRandom fail, raise an error and do not allow the user to continue. Record the error details, and encourage the user to submit the information to developers of support forums to allow diagnosing the failure.