Why is there a hash of a weak password in the Windows cryptographic libraries?

devblogs.microsoft.com/oldnewthing/20231024-00

October 24, 2023

Raymond Chen

A customer found the byte sequence `ba7816bf8f01cfea414140de5dae2223-b00361a396177a9cb410ff61f20015ad` in the Windows cryptographic libraries. This is the SHA256 hash of the notoriously insecure password `abc`. (See pages 14 through 16 of the NIST Computer Security Resource Center, Cryptographic Standards and Guidelines, SHA examples document.) Why does the Windows cryptographic library use such a ridiculously weak password, and what is this password used for?

While it's true that `abc` is a horrible password, it's also the case that the Windows cryptographic libraries aren't using it as a password. The value is part of a self-test that the libraries perform to verify that nothing obvious has gone wrong with the standard providers.

You can find this hard-coded "well-known SHA256" in the sha256.c module, with the "plaintext" in selftest.c. The values are used by the function `SymCryptSha256SelfTest` to verify that the algorithm produces the expected answer.

The fact that an insecure password appears in the cryptography libraries doesn't mean that the library is using them as passwords. In this case, they are just test data.

Bonus chatter: I bet you can find insecure passwords in a lot of binaries if you set your mind to it. Just scan for the bytes `61 62 63` in any binary, and if you find it, you can get all excited: "Hey, your binary contains the insecure password `abc`!"