Something I have

Usecases of a Yubikey

Onkobu Tanaake

2019-10-21T21:00:00

As an IT professional I have to deal with many different systems and platforms. I run some servers myself and I use different mail addresses. In addition I sign software packages as well as e-mail. There's also a lot of encryption going on. Without a keystore I'd be lost. There is a plethora of passwords, not only to access something but also to unlock encryption- or signing keys.

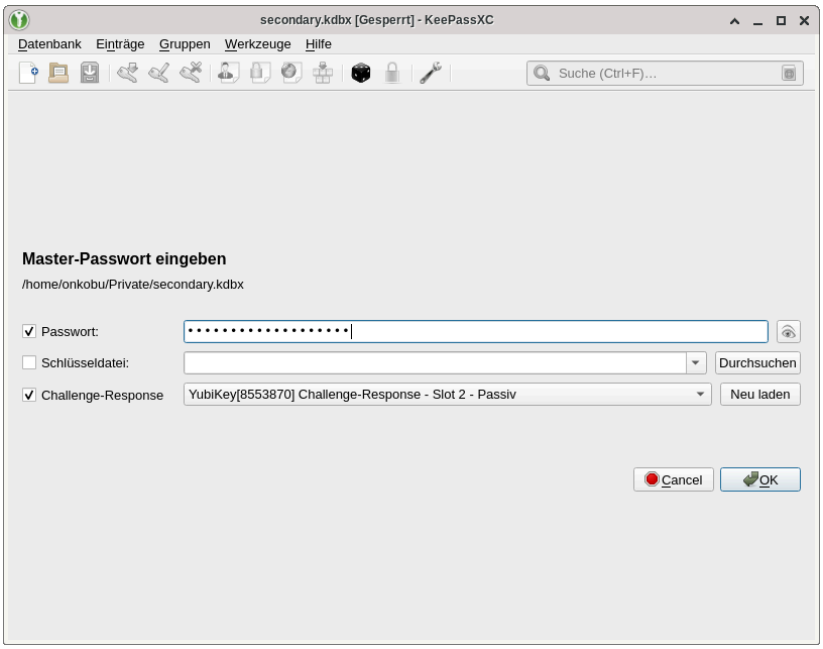

Figure 1. KeepassXC with Yubikey as 2nd factor, slot 2

Usecase #1: KeypassXC. Lately I started to use a password manager. It offers a second factor to lock/ unlock and encrypt all secrets. This HMAC-SHA1 challenge-response mechanism is in use with Slot #2 of my Yubikey. Without the Yubikey I cannot unlock any of my secrets. That is why I have to pay attention to this piece of hardware or all accounts are lost. In all other cases it is a good idea to have a second Yubikey as backup programmed with the same challenge-response parameters. It's also a good idea to backup the keystore and save it somewhere else. If it's not in at least 3 different places it doesn't exist at all. Yubikey's Personalization Tool does the configuration job very well. It can write the same properties to multiple keys. I also use separate Yubikeys for personal and professional concerns.

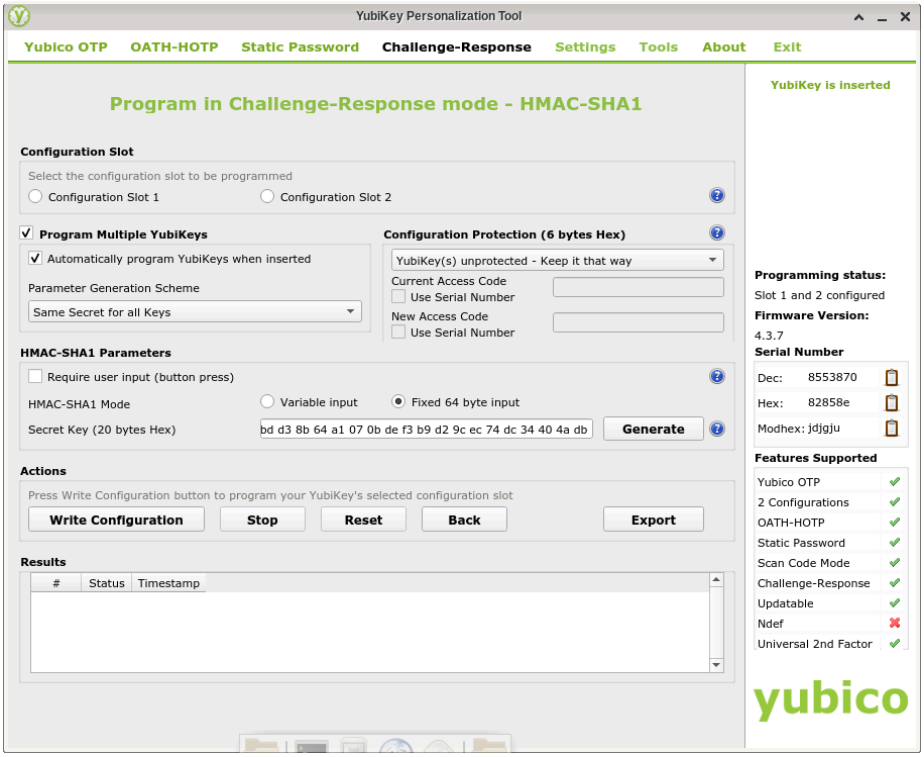

Figure 2. Personalization tool to configure HMAC-SHA1 for multiple keys with same secret

Usecase #2: SmartCard/ Cryptography. With version 5 of the Yubikey its elliptic curve capabilities were improved. I couldn't convince my version 4 key to accept any of the elliptic curve keys I created. So I fed it with 4096-bit RSA keys. Each slot is occupied by a key of its own. So there is one for authentication, one for signing and one for encryption. On Linux it is necessary to run the pcscd-daemon so that Yubikey Manager can access the CCID-interface. Despite this option I opted for gpg and its builtin SmartCard-interface. This works right away with no additional services at all. The only downside is the selection of elliptic curves. Yubico stayed for version 5 with the NIST-recommended curves ECC p256 and ECC p384.

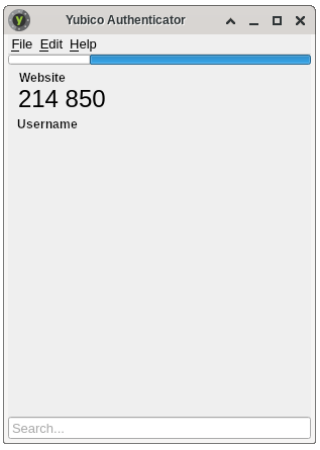

Figure 3. Authenticator displaying a secret. Double click on the digits copies to clipboard

Usecase #3: Authenticator. A very surprising finding for me was that any Yubikey can be used as generator for Authenticator keys. There is a Yubico Authenticator

application that again needs pcscd to access the CCID-interface. With the help of this application any QR-code can be turned into a OATH number generator. There is no need to run an app on an insecure mobile device. And it is even the perfect replacement for a browser plugin running in the same context as the web site requiring the number generated as second factor. With version 5 this even works with NFC into any compatible mobile device.

Usecase #4: Universal 2nd Factor/ U2F. Finally this was the first use of my Yubikey. With a simple tap gesture it emits a character sequence that can be cross checked for verification. It works for Codeberg, GitHub and many other sites. It does not send a text message onto my phone and doesn't require one at all. Pay attention that your service of choice allows adding multiple second factors to allow backup keys. In the case of GitHub it is not possible to switch off SMS. That's why I moved to Codeberg. They combine Authenticator with U2F.

Now you want to know the downsides of a Yubikey? You need at least two. As said before I even split my keys between work and private use with no overlaps. They're quite expensive, 50€ per piece must be earned first. You might be lucky in case your employer pays for it or it could be relevant for your tax declaration. The built-in counter has limited memory. But its 32bit will last 10-15 years according to Yubico. Finally some links:

- Thomas Leister's Blog [https://thomas-leister.de/yubikey-in-der-praxis/]
- Personalization Tool [https://support.yubico.com/support/solutions/articles/15000006424-yubikey-personalization-tool-user-guide]
- PCSC Lite Gentoo [https://wiki.gentoo.org/wiki/PCSC-Lite]
- Smart Cards Archwiki [https://wiki.archlinux.org/index.php/Smartcards]
- GnuPG Smartcard [https://wiki.gnupg.org/SmartCard]

Some details for the counter, which is not easy (have a look at the manual). It consists of a non-volatile and a volatile part. The first is only incremented on power up, the second upon usage. To quote from the manual: »…onsidering a YubiKey being used five times a day, 365 days per year, it will take 18 years for the counter to get stuck.« Take into consideration that the non-volatile counter is only 15bit to be compatible with version 1 keys. Therefore you should favor HMAC-SHA1 mode for challenge-response instead of counter based OTP to avoid this lifetime boundary. (Resetting the device is possible to also reset the counter but then all configuration is lost, too.)