NAME

yeps – The Yubikey Enhanced Password Safe

SYNOPSIS

yeps [-gpg] [status] | [store | read | update] identifier

DESCRIPTION

yeps safely stores cleartext secrets that the user provides encrypted both with AES and RSA. The security of yeps is based on privilege separation and on the use of a Yubikey to provide additional random characters for the conventional AES encryption of the secret to be stored. In additon to this encryption there is a second encryption with a RSA public key before the result is stored in the file system. When the secret is to be recovered, the RSA encryption has to be reverted by using the corresponding private RSA key which is stored non-exportable in the Yubikey. Following the RSA decryption the Yubikey is used to provide the additional random characters in order to revert the AES encryption. The passphrase used for conventional AES encryption can be prefixed by arbitrary user input before pressing the Yubikey button will finish the passphrase with its static password output.

OPTIONS

−gpg use gpg instead of the cryptlib based claes for conventional AES encryption

status output some information on the state of the Yubikey

store identifier
asks for a secret and stores it in the user yeps' home directory AES and RSA encrypted. The secret is limited to 60 characters.

read identifier
recovers the stored secret by reverting the RSA and AES encryption. Finally the secret is visible in the user's terminal and nowhere else.

update identifier
asks for a new secret to overwrite the data stored for the identifier

NOTES

Full documentation <https://senderek.ie/yeps>

This program depends on tree packages providing the program claes, the cryptlib shared object library and the python3-bindings to this library.

You can download all of these packages in RPM or DEB format at https://senderek.ie/cryptlib/downloads

Using FEDORA you can install the packages cryptlib, cryptlib-python3 and cryptlib-tools directly from the repository.

In addition the program /bin/systemd-ask-password is needed to read sensible data from stdin. This program is part of the systemd package.

PREPARATIONS

Dedicated user yeps

In order to ensure safe operation of writing or reading encrypted secrets, a separate user **yeps** has to be created. Its sole purpose is to shield these processes and the stored encrypted secrets from software that is running under the user's UID.

The program **yeps.py** is stored with minimal access permissions in the user yeps' home directory. When the Yubikey is being prepared (see below) a RSA keypair is being created. The public part of these RSA keys is stored in the file **/home/yeps/keys/yubi-pubkey.pem** in the encrypted keys

directory while the private RSA key remains inaccessible inside the Yubikey.

The login password of the user yeps has to be set up (and used) in the following way:

This password consists of an arbitrary number of characters (that the user can remember easily) followed by the number of random characters (the static key) that the Yubikey appends to the user's input when its button is pressed for more than 2 seconds. As the Yubikey finishes its output with a newline, the user's input (if any) has to be typed before the button is pressed.

When the secret to be protected is AES encrypted, a similar passphrase is constructed in the same way. It is left to the user whether he uses a different input for this encryption passphrase in addition with the same static key produced by the Yubikey.

YUBIKEY

It is assumed that you have changed the Yubikey's management key as well as its user access PIN see: https://developers.yubico.com/PIV/Introduction/YubiKey_and_PIV.html

The Yubikey has to provide two different features:

The static random passphrase

In order to enhance the user's input to a passphrase the Yubikey can add some 32 random characters (out of 64, ie 6 bit) when the Yubikey's button is pressed. This enhances the entropy of the password considerably without burdening the user to memorize any complex static password.

You can use **yubikey-personalization-gui** to set a fixed random passphrase in the Yubikey's slot2 configuration, which enables the device to produce this fixed passphrase at the long push of its button in addition to any user input.

see: https://docs.yubico.com/yesdk/users-manual/application-otp/static-password.html

The decryption with a RSA private key in slot 9A

With the following command **yubico-piv-tool -s9a -v -ARSA2048 -agenerate -k** you can generate a RSA keypair inside the Yubikey in slot 9A. The Yubikey's management key is needed for this operation.

The public part is then displayed after generation and the private RSA key remains inside the Yubikey. Though it cannot be exported outside the Yubikey, it still can be used to decrypt messages that have been RSA encrypted with the corresponding public key. The decryption is performed inside the Yubikey as well.

The size of the message is limited to 256 Bytes if a 2048 bit RSA key is being generated. Enough space to protect the PGP message which is the result of the conventional AES encryption of the user's secret.

FILES

/usr/bin/claes

This program is used for safe AES encryption base on Cryptlib by default. It can be replaced by gpg with the "-gpg" option.

/home/yeps/bin/yeps.py

The workhorse that is invoked via "su -l yeps" by the user running under yeps' UID.

/home/yeps/keys/yubi-pubkey.pem

The RSA public key used for RSA encryption. The corresponding RSA private key is stored in slot 9A of the Yubikey.

/home/yeps/keys
: The directory in which (double) encrypted secrets are stored in a file "identifier.rsa"

/usr/bin/systemd-ask-password
: This program is used to provide the passphrase based on a user's input. The user's input is then appended with a number of random characters released after pressing the button on the Yubikey.

/lib64/libcl.so.3.4.6
: The cryptlib library.

/usr/lib/python3.10/site-packages/cryptlib_py.so
: Bindings to the cryptlib library used by python3.

BUGS

Please report bugs to innovation@senderek.ie

AUTHORS

yeps is written by Ralf Senderek <innovation@senderek.ie>.

COPYRIGHT

SEE ALSO

claes, cryptlib, yubikey-personalization-gui, pkcs11-tool, yubico-piv-tool