NAME

pem2openpgp — translate PEM-encoded RSA keys to OpenPGP certificates

SYNOPSIS

```
pem2openpgp $USERID < mykey.pem | gpg --import

PEM2OPENPGP_EXPIRATION=$((86400 * $DAYS))
            PEM2OPENPGP_USAGE_FLAGS=authenticate,certify pem2openpgp
            $USERID <mykey.pem
```

DESCRIPTION

pem2openpgp is a low-level utility for transforming raw, PEM-encoded RSA secret keys into OpenPGP-formatted certificates. The generated certificates include the secret key material, so they should be handled carefully.

It works as an element within a pipeline: feed it the raw key on stdin, supply the desired User ID as a command line argument. Note that you may need to quote the string to ensure that it is entirely in a single argument.

Other choices about how to generate the new OpenPGP certificate are governed by environment variables.

ENVIRONMENT

The following environment variables influence the behavior of **pem2openpgp**:

PEM2OPENPGP_TIMESTAMP controls the timestamp (measured in seconds since the UNIX epoch) indicated as the creation time (a.k.a "not valid before") of the generated certificate (self-signature) and the key itself. By default, **pem2openpgp** uses the current time.

PEM2OPENPGP_KEY_TIMESTAMP controls the timestamp (measured in seconds since the UNIX epoch) indicated as the creation time of just the key itself (not the self-signature). By default, **pem2openpgp** uses the value from PEM2OPENPGP_TIMESTAMP.

PEM2OPENPGP_USAGE_FLAGS should contain a comma-separated list of valid OpenPGP usage flags (see section 5.2.3.21 of RFC 4880 for what these mean). The available choices are: certify, sign, encrypt_comms, encrypt_storage, encrypt (this means both encrypt_comms and encrypt_storage), authenticate, split, shared. By default, **pem2openpgp** only sets the certify flag.

PEM2OPENPGP_EXPIRATION sets an expiration (measured in seconds after the creation time of the key) in each self-signature packet. By default, no expiration subpacket is included.

PEM2OPENPGP_NEWKEY indicates that **pem2openpgp** should ignore stdin, and instead generate a new key internally and build the certificate based on this new key. Set this variable to the number of bits for the new RSA key (e.g. 3072). By default (when this is unset), **pem2openpgp** will read the key from stdin.

AUTHOR

pem2openpgp and this man page were written by Daniel Kahn Gillmor <dkg@fifthhorseman.net>.

BUGS

Only handles RSA keys at the moment. It might be nice to handle DSA keys as well.

Currently only creates certificates with a single User ID. Should be able to create certificates with multiple User IDs.

Currently only accepts unencrypted RSA keys. It should be able to deal with passphrase-locked key material.

Currently outputs OpenPGP certificates with cleartext secret key material. It would be good to be able to lock the output with a passphrase.

If you find other bugs, please report them at https://labs.riseup.net/code/projects/show/monkeysphere

SEE ALSO

`openpgp2ssh`(1), `monkeysphere`(1), `monkeysphere`(7), `ssh`(1), `monkeysphere-host`(8), `monkeysphere-authentication`(8)