#!/bin/bash

# email-signed-keys - script to sign and encrypt keys from the most recent BLU keysigning party, then create a batch-SMTP (BSMTP) file of email messages to send each key to the email addresses found on the key's uids

# Copyright (c) 2011 John M. Abreau <jabr@blu.org, abreauj@gmail.com>
# Released under GPL version 3 or greater - http://gplv3.fsf.org/
# usage: email-signed-keys [ --local-user signing-keyid ] keyid ... 

VERSION=0.3
GPGOPT='--quiet'
OUT_MODE=mailx

while [ $# -gt 0 ]; do
  case "$1" in
    --local-user)
      # sign with non-default secret key
      GPGOPT="$GPGOPT --local-user $2"
      shift 2
      ;;
    --bsmtp)
      OUT_MODE=bsmtp
      ;;
    --mailx)
      OUT_MODE=mailx
      ;;
    --)
      shift; break ;;
    -*)
      shift ;
      *)
      break ;;
  esac
done

# get name and email from first uid of signing key
FROM=$(gpg $GPGOPT --list-secret | grep ^uid | head -1 | sed 's/^uid  *///')

# get date of most recent BLU keysigning party
DATE=$(date +%F --date "$(curl -s http://blu.org/keysignings/ | \ grep ' :: ' | head -1 | sed 's/ :: .*//')")

case "$OUT_MODE" in
  bsmtp)
    OUT=bsmtp.$DATE.shar
    ;;
  mailx)
    OUT=mailx.$DATE.shar
    mailx=$(type -p mailx)
    [ "$mailx" = "" ] && mailx=$(type -p Mail)
    [ "$mailx" = "" ] && mailx=/bin/mail
    ;;
    *)
    OUT=bsmtp.$DATE.shar
    ;;
  esac

LONGDATE=$(date '+%A, %B %d, %Y' --date $DATE)
SENDER=${FROM#*<}
SENDER=${SENDER%>*}
SENDERNAME=${FROM% <*}

main() {
  echo '#! /bin/sh' > $OUT
  write_tmpl
  for keyid do

echo '###' Keyid $keyid '###'
sign-and-encrypt $keyid
gpg $GPGOPT --list-key $keyid | \
grep '^uid' | \
sed -e 's/>.*/'' -e 's/>.*//' | \
while read addr ; do
    make_message $keyid $addr >> $OUT
done
echo 'exit 0' >> $OUT
}
sign-and-encrypt() {
    local keyid=$1
gpg $GPGOPT --recv-key $keyid
gpg $GPGOPT --sign-key $keyid
    [ −e $keyid−exported.asc ] || gpg $GPGOPT --export −a −o $keyid−exported.asc
    [ −e $keyid−signed.asc ] || gpg $GPGOPT --yes −sear $keyid −o $keyid−signed.asc
}
make_message() {
    local keyid=$1
    local addr=$2

case "$OUT_MODE" in
    bsmtp) make_bsmtp_header "$keyid" "$addr" ;;
    mailx) make_mailx_header "$keyid" "$addr" ;;
    *) make_mailx_header "$keyid" "$addr" ;;
esac
echo ''
cat TMPL | sed \n    -e "s/%{addr}$/addr/g" \n    -e "s/%{date}$/DATE/g" \n    -e "s/%{longdate}$/LONGDATE/g" \n    -e "s/%{keyid}$/keyid/g" \n    -e "s/%{sender_addr}$/SENDER/g" \n    -e "s/%{sender_name}$/SENDERNAME/g"
echo ''
cat $keyid−signed.asc

echo .
echo SHAR_EOF
}
make_bsmtp_header() {
    local keyid=$1
    local addr=$2

echo '/usr/sbin/sendmail −v −bs << \SHAR_EOF'
    echo HELO localhost
    echo "MAIL FROM: <$SENDER>
    echo "RCPT TO: <$addr>"
    echo DATA
    echo Subject: Signed, encrypted key $keyid from BLU keysigning $DATE
    echo From: "$SENDERNAME <$SENDER>"
    echo To: $addr
    echo X−Mailer: BLU Keysigning Script $VERSION
}
make_mailx_header() {
    local keyid=$1
    local addr=$2

Below is the key %{keyid} from the BLU keysigning party that was held on %{longdate} at M.I.T. This key was signed by %{sender_name} <%x{sender_addr}>, extracted from his/her keyring, and encrypted so that only the person with the secret key %{keyid} can read it. This message is sent to

    %{addr}

which is the address (or one of the addresses) contained within the key %{keyid}. Only the legitimate owner of that key can read the signature.

Once you decrypt the message, you should import the signed key with 'gpg --import', then upload the key to the keyservers with 'gpg --send-keys %{keyid} --keyserver subkeys.pgp.net'.