:

Astra Linux Special Edition .10015-01 (1.7), .10015-10
Astra Linux Special Edition .10015-17
Astra Linux Special Edition .10015-37 (7.7)
Astra Linux Special Edition .10015-03 (7.6)
Astra Linux Special Edition .10152-02 (4.7)
Astra Linux Special Edition .10015-01 (1.6)
Astra Linux Special Edition .10015-16 . 1
Astra Linux Special Edition .10015-16 . 2
Astra Linux Special Edition .10265-01 (8.1)
Astra Linux Common Edition 2.12

Astra Linux Special Edition 1.6, 8.1, x.7

:

- bsign;
- binutils;
- gzip;
- lzma;
- bzip2;
- gnupg.

:

```
sudo apt install bsign binutils gzip lzma bzip2 gnupg
```

1. :

   ```
   sudo gpg --import secret_key.key
   ```

2. :

   ```
   sudo gpg --list-secret-keys
   ```

1. sudo.
2. sign-deb.sh. web- sign-deb16.sh.
3. :

   ```
   key_id="<_>"
   pass_file="<_____>"
   ```

4. :

   a. , , ;
   b. , , :

   ```
   sudo ./sign-deb16.sh /root/pkg /root/pkg_sign
   ```

:

sign-deb16.sh

```
#!/bin/bash
#---------------------
key_id=""
pass_file=""
DIR_BIN=$1
DIR_SIGNED=$2
#---------------------
TMP_DIR=$DIR_SIGNED/tmp$$

if [ -z $DIR_BIN ] ; then
    echo "Specify original directory as first argument."
    exit 1
fi

if [ -z $DIR_SIGNED ] ; then
    echo "Specify destination directory as second argument."
    exit 1
fi

if [ ! -e $DIR_BIN ] ; then
    echo "Original directory $DIR_BIN doesn't exist."
    exit 1
fi

list_of_packages=`find $DIR_BIN -type f -name "*.deb"`
list_of_udebs=`find $DIR_BIN -type f -name "*.udeb"`

for i in $list_of_packages ; do
    pack_name=`echo $i | awk '{sub(/^.+\//,"",$0) ; a=$0; print a}'`
    echo "Unpacking $pack_name"
    mkdir -p $TMP_DIR/{control,data}
    cp $i $TMP_DIR
    pushd $TMP_DIR &>/dev/null
    fakeroot ar x $pack_name
    SIGN_LOG=${DIR_SIGNED}/${pack_name}.log
# Definig archives type
    data_arch_type=`ls data.tar* | cut -d'.' -f3`
    control_arch_type=`ls control.tar* | cut -d'.' -f3`
    popd &>/dev/null
# Unpack data archive
    pushd $TMP_DIR/data &>/dev/null
    case $data_arch_type in
        gz)
        fakeroot tar --same-permissions --same-owner -xzf ../data.tar.gz
        ;;
        bz2)
        fakeroot tar --same-permissions --same-owner -xjf ../data.tar.bz2
        ;;
        lzma)
        fakeroot tar --same-permissions --same-owner --lzma -xf ../data.tar.lzma
        ;;
        xz)
        fakeroot tar --same-permissions --same-owner -xJf ../data.tar.xz
        ;;
    esac
    popd &>/dev/null

# Unpack control archive
    pushd $TMP_DIR/control &>/dev/null
    case $control_arch_type in
        gz)
        fakeroot tar --same-permissions --same-owner -xzf ../control.tar.gz
        ;;
        bz2)
        fakeroot tar --same-permissions --same-owner -xjf ../control.tar.bz2
```

```
        ;;
        lzma)
        fakeroot tar --same-permissions --same-owner --lzma -xf ../control.tar.lzma
        ;;
        xz)
        fakeroot tar --same-permissions --same-owner -xJf ../control.tar.xz
        ;;
    esac
    popd &>/dev/null

# Sign files
    pushd $TMP_DIR/data &> /dev/null
    list=`find . -type f`
    for file in $list ; do
        elf=`file $file | cut -d: -f2 | awk -F' ' '{print $1}'`
        if [ $elf = "ELF" ]; then
        #if [ "$(file $file | grep ELF)" ];then
            oldstat=`stat -c %a $file`
            bsign -N -s --pgoptions="--batch --pinentry-mode=loopback --passphrase-file=$pass_file --default-
key=$key_id" $file
            bsign -w $file
            newstat=`stat -c %a $file`
            [ $newstat != $oldstat ] && echo "BSIGN_CHMOD_ERROR in $file" >> ${SIGN_LOG} 2>&1
        else
            echo "$file not ELF"
        fi
    done
    popd &> /dev/null

# Counting md5sums
    pushd $TMP_DIR/control &>/dev/null
    if [ -e ./md5sums ] ; then
        filenames=`cat md5sums | awk -F'  ' '{print $2}'`
        popd &>/dev/null
        pushd $TMP_DIR/data &> /dev/null
        for j in $filenames
        do
            echo `md5sum $j` >> $TMP_DIR/control/md5sums.new
        done
        sed -e 's/\ /\ \ /g' $TMP_DIR/control/md5sums.new > $TMP_DIR/control/md5sums.new_mod
        popd &> /dev/null
        mv -f $TMP_DIR/control/md5sums.new_mod $TMP_DIR/control/md5sums &> /dev/null
        rm -f $TMP_DIR/control/md5sums.new &> /dev/null
    fi

# Packing back in deb
    pushd $TMP_DIR/data &> /dev/null
    case $data_arch_type in
        gz)
        fakeroot tar --same-permissions --same-owner -czf ../data.tar.gz .
        ;;
        bz2)
        fakeroot tar --same-permissions --same-owner -cjf ../data.tar.bz2 .
        ;;
        lzma)
        fakeroot tar --same-permissions --same-owner --lzma -cf ../data.tar.lzma .
        ;;
        xz)
        fakeroot tar --same-permissions --same-owner -cJf ../data.tar.xz .
        ;;
    esac
    popd &> /dev/null

    pushd $TMP_DIR/control &> /dev/null
    case $control_arch_type in
        gz)
        fakeroot tar --same-permissions --same-owner -czvf ../control.tar.gz .
        ;;
        bz2)
        fakeroot tar --same-permissions --same-owner -cjvf ../control.tar.bz2 .
        ;;
```

```
        lzma)
        fakeroot tar --same-permissions --same-owner --lzma -cvf ../control.tar.lzma .
        ;;
        xz)
        fakeroot tar --same-permissions --same-owner -cJvf ../control.tar.xz .
        ;;
    esac
    popd &> /dev/null
    pushd $TMP_DIR &> /dev/null
    fakeroot ar rcs $TMP_DIR/$pack_name debian-binary control.tar.$control_arch_type data.
tar.$data_arch_type &> /dev/null
    cp ${pack_name}  $DIR_SIGNED/${pack_name%.deb}_signed.deb &> /dev/null
    echo "Creating $DIR_SIGNED/${pack_name%.deb}_signed.deb"
    popd &> /dev/null
    rm -rf $TMP_DIR  &> /dev/null
done

for j in $list_of_udebs ; do
    cp $j $DIR_SIGNED
done

echo "done"
exit 0
```

, Astra Linux Special Edition 1.5

, , Astra Linux Special Edition 1.5 , astra-digsig-oldkeys. /etc/digsig/keys/legacy/keys/

Astra Linux Special Edition .10015.01 1.4 1.5

sudo. :

```
sudo gpg --import key_for_signing.key
```

. root. , , . , .

sign_deb.sh

```
#!/bin/bash

key_id="00000000"
pass_file="/root/key_password.txt"

DIR_BIN=$1
DIR_SIGNED=$2

TMP_DIR=$DIR_SIGNED/tmp$$
SIGN_LOG=sign.log

if [ -z $DIR_BIN ] ; then
        echo "Specify original directory as first argument."
        exit 1
fi

if [ -z $DIR_SIGNED ] ; then
        echo "Specify destination directory as second argument."
        exit 1
fi

if [ ! -e $DIR_BIN ] ; then
        echo "Original directory $DIR_BIN doesn't exist."
        exit 1
```

```
fi

list_of_packages=`find $DIR_BIN -type f -name "*.deb"`
list_of_udebs=`find $DIR_BIN -type f -name "*.udeb"`

for i in $list_of_packages ; do
    pack_name=`echo $i | awk '{sub(/^.+\//,"",$0) ; a=$0; print a}'`
    echo "Unpacking $pack_name"
    mkdir -p $TMP_DIR/{control,data}
    cp $i $TMP_DIR
    pushd $TMP_DIR &>/dev/null
    ar x $pack_name
    SIGN_LOG=${DIR_SIGNED}/${pack_name}.log
# Definig archives type
    data_arch_type=`ls data.tar* | cut -d'.' -f3`
    control_arch_type=`ls control.tar* | cut -d'.' -f3`
    popd &>/dev/null
# Unpack data archive
    pushd $TMP_DIR/data &>/dev/null
    case $data_arch_type in
        gz)
        tar --same-permissions --same-owner -xzf ../data.tar.gz
        ;;
        bz2)
        tar --same-permissions --same-owner -xjf ../data.tar.bz2
        ;;
        lzma)
        tar --same-permissions --same-owner --lzma -xf ../data.tar.lzma
        ;;
        xz)
        tar --same-permissions --same-owner -xJf ../data.tar.xz
        ;;
    esac
    popd &>/dev/null

# Unpack control archive
    pushd $TMP_DIR/control &>/dev/null
    case $control_arch_type in
        gz)
        tar --same-permissions --same-owner -xzf ../control.tar.gz
        ;;
        bz2)
        tar --same-permissions --same-owner -xjf ../control.tar.bz2
        ;;
        lzma)
        tar --same-permissions --same-owner --lzma -xf ../control.tar.lzma
        ;;
        xz)
        tar --same-permissions --same-owner -xJf ../control.tar.xz
        ;;
    esac
    popd &>/dev/null

# Sign files
    pushd $TMP_DIR/data &> /dev/null
    list=`find . -type f`
    for file in $list ; do
        elf=`file $file | cut -d: -f2 | awk -F' ' '{print $1}'`
        if [ $elf = "ELF" ]; then
        #if [ "$(file $file | grep ELF)" ];then
            oldstat=`stat -c %a $file`
            bsign -N -s --pgoptions="--default-key=$key_id --passphrase-file=$pass_file" $file
            bsign -w $file
            newstat=`stat -c %a $file`
            [ $newstat != $oldstat ] && echo "BSIGN_CHMOD_ERROR in $file" >> ${SIGN_LOG} 2>&1
        else
            echo "$file not ELF"
        fi
    done
    popd &> /dev/null
```

```
# Counting md5sums
    pushd $TMP_DIR/control &>/dev/null
    if [ -e ./md5sums ] ; then
        filenames=`cat md5sums | awk -F'  ' '{print $2}'`
        popd &>/dev/null
        pushd $TMP_DIR/data &> /dev/null
        for j in $filenames
        do
            echo `md5sum $j` >> $TMP_DIR/control/md5sums.new
        done
        sed -e 's/\ /\ \ /g' $TMP_DIR/control/md5sums.new > $TMP_DIR/control/md5sums.new_mod
        popd &> /dev/null
        mv -f $TMP_DIR/control/md5sums.new_mod $TMP_DIR/control/md5sums &> /dev/null
        rm -f $TMP_DIR/control/md5sums.new &> /dev/null
    fi

# Packing back in deb
    pushd $TMP_DIR/data &> /dev/null
    case $data_arch_type in
        gz)
        tar --same-permissions --same-owner -czf ../data.tar.gz .
        ;;
        bz2)
        tar --same-permissions --same-owner -cjf ../data.tar.bz2 .
        ;;
        lzma)
        tar --same-permissions --same-owner --lzma -cf ../data.tar.lzma .
        ;;
        xz)
        tar --same-permissions --same-owner -cJf ../data.tar.xz .
        ;;
    esac
    popd &> /dev/null

    pushd $TMP_DIR/control &> /dev/null
    case $control_arch_type in
        gz)
        tar --same-permissions --same-owner -czvf ../control.tar.gz .
        ;;
        bz2)
        tar --same-permissions --same-owner -cjvf ../control.tar.bz2 .
        ;;
        lzma)
        tar --same-permissions --same-owner --lzma -cvf ../control.tar.lzma .
        ;;
        xz)
        tar --same-permissions --same-owner -cJvf ../control.tar.xz .
        ;;
    esac
    popd &> /dev/null
    pushd $TMP_DIR &> /dev/null
    ar rcs $TMP_DIR/$pack_name debian-binary control.tar.$control_arch_type data.tar.$data_arch_type &> /dev
/null
    cp ${pack_name}  $DIR_SIGNED/${pack_name%.deb}_signed.deb &> /dev/null
    echo "Creating $DIR_SIGNED/${pack_name%.deb}_signed.deb"
    popd &> /dev/null
    rm -rf $TMP_DIR  &> /dev/null
done

for j in $list_of_udebs ; do
    cp $j $DIR_SIGNED
done

echo "Sign done"
```

:

```
key_id="000000"
pass_file="/root/key_password.txt"
```

00000 , , /root/key_password.txt . , :

```
gpg --list-keys
```

sign-deb_package.sh

Astra Linux Special Edition .10015-01 1.5 ,

sudo. :

```
sudo gpg --import key_for_signing.key
```

key_for_signing.key — .

. root. , , . , .

sign-deb.sh

```
#!/bin/bash

key_id="00000000"
pass_file="/root/key_password.txt"

DIR_BIN=$1
DIR_SIGNED=$2

TMP_DIR=$DIR_SIGNED/tmp$$
SIGN_LOG=sign.log

if [ -z $DIR_BIN ] ; then
        echo "Specify original directory as first argument."
        exit 1
fi

if [ -z $DIR_SIGNED ] ; then
        echo "Specify destination directory as second argument."
        exit 1
fi

if [ ! -e $DIR_BIN ] ; then
        echo "Original directory $DIR_BIN doesn't exist."
        exit 1
fi

list_of_packages=`find $DIR_BIN -type f -name "*.deb"`
list_of_udebs=`find $DIR_BIN -type f -name "*.udeb"`

for i in $list_of_packages ; do
    pack_name=`echo $i | awk '{sub(/^.+\//,"",$0) ; a=$0; print a}'`
    echo "Unpacking $pack_name"
    mkdir -p $TMP_DIR/{control,data}
    cp $i $TMP_DIR
    pushd $TMP_DIR &>/dev/null
    ar x $pack_name
    SIGN_LOG=${DIR_SIGNED}/${pack_name}.log
# Definig archives type
    data_arch_type=`ls data.tar* | cut -d'.' -f3`
    control_arch_type=`ls control.tar* | cut -d'.' -f3`
    popd &>/dev/null
# Unpack data archive
    pushd $TMP_DIR/data &>/dev/null
    case $data_arch_type in
        gz)
        tar --same-permissions --same-owner -xzf ../data.tar.gz
```

```
        ;;
        bz2)
        tar --same-permissions --same-owner -xjf ../data.tar.bz2
        ;;
        lzma)
        tar --same-permissions --same-owner --lzma -xf ../data.tar.lzma
        ;;
        xz)
        tar --same-permissions --same-owner -xJf ../data.tar.xz
        ;;
    esac
    popd &>/dev/null

# Unpack control archive
    pushd $TMP_DIR/control &>/dev/null
    case $control_arch_type in
        gz)
        tar --same-permissions --same-owner -xzf ../control.tar.gz
        ;;
        bz2)
        tar --same-permissions --same-owner -xjf ../control.tar.bz2
        ;;
        lzma)
        tar --same-permissions --same-owner --lzma -xf ../control.tar.lzma
        ;;
        xz)
        tar --same-permissions --same-owner -xJf ../control.tar.xz
        ;;
    esac
    popd &>/dev/null

# Sign files
    pushd $TMP_DIR/data &> /dev/null
    list=`find . -type f`
    for file in $list ; do
        elf=`file $file | cut -d: -f2 | awk -F' ' '{print $1}'`
        if [ $elf = "ELF" ]; then
        #if [ "$(file $file | grep ELF)" ];then
            oldstat=`stat -c %a $file`
            bsign -9 -N -s --pgoptions="--default-key=$key_id --passphrase-file=$pass_file" $file
            bsign -w $file
            newstat=`stat -c %a $file`
            [ $newstat != $oldstat ] && echo "BSIGN_CHMOD_ERROR in $file" >> ${SIGN_LOG} 2>&1
        else
            echo "$file not ELF"
        fi
    done
    popd &> /dev/null

# Counting md5sums
    pushd $TMP_DIR/control &>/dev/null
    if [ -e ./md5sums ] ; then
        filenames=`cat md5sums | awk -F'  ' '{print $2}'`
        popd &>/dev/null
        pushd $TMP_DIR/data &> /dev/null
        for j in $filenames
        do
            echo `md5sum $j` >> $TMP_DIR/control/md5sums.new
        done
        sed -e 's/\ /\ \ /g' $TMP_DIR/control/md5sums.new > $TMP_DIR/control/md5sums.new_mod
        popd &> /dev/null
        mv -f $TMP_DIR/control/md5sums.new_mod $TMP_DIR/control/md5sums &> /dev/null
        rm -f $TMP_DIR/control/md5sums.new &> /dev/null
    fi

# Packing back in deb
    pushd $TMP_DIR/data &> /dev/null
    case $data_arch_type in
        gz)
        tar --same-permissions --same-owner -czf ../data.tar.gz .
        ;;
```

```
        bz2)
        tar --same-permissions --same-owner -cjf ../data.tar.bz2 .
        ;;
        lzma)
        tar --same-permissions --same-owner --lzma -cf ../data.tar.lzma .
        ;;
        xz)
        tar --same-permissions --same-owner -cJf ../data.tar.xz .
        ;;
    esac
    popd &> /dev/null

    pushd $TMP_DIR/control &> /dev/null
    case $control_arch_type in
        gz)
        tar --same-permissions --same-owner -czvf ../control.tar.gz .
        ;;
        bz2)
        tar --same-permissions --same-owner -cjvf ../control.tar.bz2 .
        ;;
        lzma)
        tar --same-permissions --same-owner --lzma -cvf ../control.tar.lzma .
        ;;
        xz)
        tar --same-permissions --same-owner -cJvf ../control.tar.xz .
        ;;
    esac
    popd &> /dev/null
    pushd $TMP_DIR &> /dev/null
    ar rcs $TMP_DIR/$pack_name debian-binary control.tar.$control_arch_type data.tar.$data_arch_type &> /dev
/null
    cp ${pack_name}  $DIR_SIGNED/${pack_name%.deb}_signed.deb &> /dev/null
    echo "Creating $DIR_SIGNED/${pack_name%.deb}_signed.deb"
    popd &> /dev/null
    rm -rf $TMP_DIR  &> /dev/null
done

for j in $list_of_udebs ; do
    cp $j $DIR_SIGNED
done

echo "Sign done"
```

:

```
key_id="000000"
pass_file="/root/key_password.txt"
```

00000 , , /root/key_password.txt - . , :

```
gpg --list-keys
```

.

GPG

GnuPG
The internals of an OpenPGP key