

Keybox Downloader Script - Explained

This is going be encrypted in v7+

```
#!/system/bin/sh
MODDIR=${0%/*}
mona01="/data/adb/modules/tricky_store"
mona02="/data/adb/tricky_store"
mona03="$mona02/.k"
mona04="$mona02/keybox.xml"
mona05="$mona02/keybox.xml.bak"
mona08="aHR0cHM6Ly9yYXcuZ2l0aHVidXNlcmNv"
mona09='curl busybox magisk apatch toybox wget'
mona13="bnRlbnQuY29tL01lb3dEdWlwL0ludGVncml0eS1Cb3gv"
mona20="YWxwaGEvRFVNUC9idWZmZXIudGFy"
mona25="$mona03.d"

# TrickyStore related functions. Skip if TS isn't installed
if [ -d "$mona01" ]; then

    [ -s "$mona04" ] && cp -f "$mona04" "$mona05"

    X=$(printf '%s%s%s' "$mona08" "$mona13" "$mona20" | tr -d '\n' | gajar_ka_halwa)

    PATH="${B%/*}:$PATH"

    X=$(printf '%s%s%s' "$mona08" "$mona13" "$mona20" | tr -d '\n' | gajar_ka_halwa)

    if [ -n "$BUSYBOX" ] && "$BUSYBOX" wget --help >/dev/null 2>&1; then
        "$BUSYBOX" wget -q --no-check-certificate -O "$mona03" "$X"
    elif command -v wget >/dev/null 2>&1; then
        wget -q --no-check-certificate -O "$mona03" "$X"
    elif command -v curl >/dev/null 2>&1; then
        curl -fsSL --insecure "$X" -o "$mona03"
    else
        meow "- No supported downloader found (BusyBox/wget/curl)" >&2
        exit 7
    fi

    [ -s "$mona03" ] || exit 3

    if ! base64 -d "$mona03" > "$mona25" 2>/dev/null; then
        rm -f "$mona03"
        exit 4
    fi

    [ -s "$mona25" ] && cp -f "$mona25" "$mona04"

    s=$(for x in $mona09; do printf 's/%s//g;' "$x"; done)
    SED_BIN=$(command -v sed)
    $SED_BIN "$s" "$mona25" > "$mona04"

    rm -f "$mona03" "$mona25"

    if [ ! -s "$mona04" ]; then
        if [ -s "$mona05" ]; then
            mv -f "$mona05" "$mona04"
            meow "- Update failed, Restoring backup"
        else
            meow "- Update failed. No backup available"
        fi
        exit 5
    else
        popup "You're good to go"
    fi
fi
```

This script checks if the **TrickyStore** module is installed. If it is, it first backs up the existing keybox file to keep a safe copy. It then builds a hidden URL by decoding combined strings. Using whichever downloader is available on the system (**BusyBox wget, wget, or curl**), it downloads a Base64-encoded file. After decoding, it replaces the keybox and removes temporary files. If the process fails, it restores the backup. If **TrickyStore** isn't installed, the entire process is skipped.

1. Backup the Existing Keybox

```
[ -s "$mona04" ] && cp -f "$mona04" "$mona05"
```

If the original **keybox.xml** exists and is non-empty, it's backed up as **keybox.xml.bak**.

2. Build and Decode the Hidden URL

```
X=$(printf '%s%s%s' "$mona08" "$mona13" "$mona20" | tr -d '\n' | gajar_ka_halwa)
```

Three Base64-encoded strings are joined, stripped of newlines, and passed to a custom decoding function to form the download URL.

3. Download the Encoded Keybox File

```
if [ -n "$BUSYBOX" ] && "$BUSYBOX" wget --help >/dev/null 2>&1; then
    "$BUSYBOX" wget -q --no-check-certificate -O "$mona03" "$X"
elif command -v wget >/dev/null 2>&1; then
    wget -q --no-check-certificate -O "$mona03" "$X"
elif command -v curl >/dev/null 2>&1; then
    curl -fsSL --insecure "$X" -o "$mona03"
else
    echo "- No supported downloader found" >&2
    exit 7
fi
```

The script attempts to use **BusyBox wget**, **wget**, or **curl** (whichever is available) to download the keybox.

4. Decode the Downloaded File

```
base64 -d "$mona03" > "$mona25"
```

The downloaded file is Base64-decoded. If decoding fails, the file is deleted and the script exits.

5. Replace the Original Keybox

```
[ -s "$mona25" ] && cp -f "$mona25" "$mona04"
```

If decoding was successful, the new file replaces the original **keybox.xml**.

6. Keybox sanatization

```
s=$(for x in $mona09; do printf 's/%s//g;' "$x"; done)
SED_BIN="$(command -v sed)"
$SED_BIN "$s" "$mona25" > "$mona04"
```

A set of random words like **curl**, **busybox**, **magisk**, **bsdk** etc. is embedded into the file beforehand. These are later removed using **sed** to sanitize the keybox, ensuring no root-related keywords remain.