# Verify a File Using an ASC Signature File

## 1. Overview

In this article, we'll learn how to verify a file using an ASC signature file using the gpg (GnuPG) command. OpenPGP (PGP) is a standard defining encryption and verification of files. GnuPG is an open-source implementation of these standards and is what we'll be using in this tutorial.

## 2. Verifying File Using ASC Signature File

Let's begin by going over some basic terms. A digital signature is a way to validate the contents of a file using the sender's public key. The digital signature may either be attached to the message or in a separate file (Detached Signature).

An ASC signature file is contained in an ASCII-Armored file and uses the .asc file extension. ASCII-armor puts binary data into ASCII and adds headers that provide information about the data inside. An ASCII-Armored PGP key file uses the .asc extension as well.

## 2.1. Adding Public Key to Keyring

To get started, we'll first need to ensure that we have the issuer's public key in our keyring. Information on where to acquire the issuer's public key will usually be in the same place as the target file. For the most part, this will also include the key's fingerprint to check against when importing.

If the public key is provided through a file, we can simply download the file and then import it:

```
gpg --show-keys --with-fingerprint publickeyfile.asc5
pub   rsa4096 2021-07-16 [SC] [expires: 2031-07-14]
      A21F AB74 B008 8AA3 6115  2586 B8EF 1A6B A9DA 2D5C
...
```

The previous command allows us to obtain the fingerprint of the key file. **It's important to compare this fingerprint to the fingerprint listed at a trusted source, such as the issuer's website, to verify that it is the correct public key**. After verifying our public key file, we can then import it:

```
gpg --import publickeyfile.asc
gpg: key B8EF1A6BA9DA2D5C: public key "Marko Oldenburg
<marko.oldenburg@cooltux.net>" imported
gpg: Total number processed: 1
gpg:               imported: 1
```

A public key may be hosted on a keyserver as well. **To import a key from a keyserver, we should use the full fingerprint of the public key.** This is important because key IDs are vulnerable to collision attacks.

```
gpg --keyserver 'keys.openpgp.org' --recv-keys
'A21FAB74B0088AA361152586B8EF1A6BA9DA2D5C'
gpg: key B8EF1A6BA9DA2D5C: public key "Marko Oldenburg
<marko.oldenburg@cooltux.net>" imported
gpg: Total number processed: 1
gpg:               imported: 1
```

To import a key from a keyserver, we use gpg with the –keyserver option passing it to the keyserver we want to search. **Since most key servers share keys with each other, it typically will not matter which keyserver we use long as it is reputable.** Then, we use the –recv-keys option with the fingerprint of the public key we want to import. The key will automatically be downloaded to our keyring if found on the key server.

## 2.2. Verifying Our File

After ensuring that we have the correct public key, we can begin verifying our file:

```
gpg --verify openssl-3.0.7.tar.gz.asc openssl-3.0.7.tar.gz
...
gpg:                using RSA key DC7032662AF885E2F47F243F527466A21CA79E6D
gpg:                issuer "marko.oldenburg@cooltux.net"
gpg: Good signature from "Marko Oldenburg <marko.oldenburg@cooltux.net>"
[unknown]
...
```

Here we use gpg –verify with our signature file and the file we want to verify. The output indicates that our file was verified successfully. The output may also contain this warning:

```
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the
owner.
```

If we have added our public key to our keyring but not to our list of trusted keys, this will be included in the output.

**In general, this warning can be safely ignored if we have properly verified that the public key in our keyring is correct.** However, if we want to prevent this warning from being displayed, we can add the associated public key to our trusted keys.

When the verification process has failed, the output includes this line:

```
gpg: BAD signature from "Marko Oldenburg <marko.oldenburg@cooltux.net>"
[unknown]
```

This output indicates our verification was unsuccessful, and we should redownload the file and signature.

From:
https://blog.cooltux.net/ - **TuxNet DokuWiki**

Permanent link:
**https://blog.cooltux.net/doku.php?id=it-wiki:linux:verify-file-asc-signature**

Last update: **2024/12/03 07:07**