## Tooling for verification of PGP signed commits

This is an incomplete work in progress, but currently includes a pre-push hook script ( `pre-push-hook.sh` ) for maintainers to ensure that their own commits are PGP signed (nearly always merge commits), as well as a Python 3 script to verify commits against a trusted keys list.

## Using verify-commits.py safely

Remember that you can't use an untrusted script to verify itself. This means that checking out code, then running `verify-commits.py` against `HEAD` is *not* safe, because the version of `verify-commits.py` that you just ran could be backdoored. Instead, you need to use a trusted version of verify-commits prior to checkout to make sure you're checking out only code signed by trusted keys:

```
git fetch origin && \
./contrib/verify-commits/verify-commits.py origin/master && \
git checkout origin/master
```

Note that the above isn't a good UI/UX yet, and needs significant improvements to make it more convenient and reduce the chance of errors; pull-reqs improving this process would be much appreciated.

## Configuration files

- `trusted-git-root` : This file should contain a single git commit hash which is the first unsigned git commit (hence it is the "root of trust").
- `trusted-sha512-root-commit` : This file should contain a single git commit hash which is the first commit without a SHA512 root commitment.
- `trusted-keys` : This file should contain a \n-delimited list of all PGP fingerprints of authorized commit signers (primary, not subkeys).
- `allow-revsig-commits` : This file should contain a \n-delimited list of git commit hashes. See next section for more info.

## Import trusted keys

In order to check the commit signatures, you must add the trusted PGP keys to your machine. GnuPG may be used to import the trusted keys by running the following command:

```
gpg --keyserver hkps://keys.openpgp.org --recv-keys $(<contrib/verify-commits/trusted-keys)
```

## Key expiry/revocation

When a key (or subkey) which has signed old commits expires or is revoked, verify-commits will start failing to verify all commits which were signed by said key. In order to avoid bumping the root-of-trust `trusted-git-root` file, individual commits which were signed by such a key can be added to the `allow-revsig-commits` file. That way, the PGP signatures are still verified but no new commits can be signed by any expired/revoked key. To easily build a list of commits which need to be added, verify-commits.py can be edited to test each commit with BITCOIN_VERIFY_COMMITS_ALLOW_REVSIG set to both 1 and 0, and those which need it set to 1 printed.