The Hitchhiker's Guide to Online Anonymity

Warning:

The public keys outlined in this document have been compromised. This document still needs to be updated to reflect the new verification process.

How to check the files for safety/integrity and authenticity.

The PDF and ODT files in this guide are cryptographically signed using GPG and Minisign. Their integrity can be verified with the published SHA256 Chrecksum Hashes on this website.

SHA256 Checksums of all the PDF and ODT files are available here in the `sha256sum.txt` file.

SHA256 Checksums, signatures, and virustotal checks of the releases files (containing the whole repository) are available within release information at `https://github.com/AnonymousPlanet/thgtoa/releases/latest`

The GPG signatures for each PDF and ODT files are available here: - PDF (Light Theme) Main and Mirrors: `guide.pdf.asc` - PDF (Dark Theme) Main and Mirrors: `guide-dark.pdf.asc` - ODT Main and Mirrors: `guide.odt.asc`

The Minisign signatures for each PDF and ODT files are available here: - PDF (Light Theme) Main and Mirrors: `guide.pdf.minisig` - PDF (Dark Theme) Main and Mirrors: `guide-dark.pdf.minisig` - ODT Main and Mirrors: `guide.odt.minisig`

How to check the integrity of the files using the SHA256 Checksums:

Please do the following:

Windows: - From a command prompt, run `certutil -hashfile filename.txt sha256` - Compare the result with the hash in the online checksum files. They should match.

MacOS: - From a terminal, run `shasum -a 256 /full/path/to/your/file` - Compare the result with the hash in the online checksum files. They should match.

Linux: - From a terminal, run `sha256sum /full/path/to/your/file` - Compare the result with the hash in the online checksum files. They should match.

All commits and releases on this repository are cryptographically signed and verified using the same GPG key. Check for the "Verified" tags on each commit or release.

How to verify the the authenticity and integrity of the files using GPG:

Now to verify the files with GPG signatures, you should first install gpg on your system: - Windows: Install gpg4win from `https://www.gpg4win.org/download.html` - MacOS: Install GPG Tools from `https://gpgtools.org/` - Linux: gpg should be installed by default

Import the GPG key using the following command from a command prompt or terminal:

```
gpg --auto-key-locate nodefault,wkd --locate-keys 0xEB16B6AB4AB7BA61F33E2DFD0051E9A5
```

In theory this command should fetch the key from the a default pool server. If this doesn't work, you can also download/view it directly from here: `https://anonymousplanet.org/AnonymousPlanet_0x89DAB601_public.asc` [[Mirror]][12] [[Tor Mirror]][14]

For redundancy, you can also verify the authenticity of this GPG signature using: - My Keybase.io profile `https://keybase.io/anonymousplanet` - My Keyoxide.org profile `https://keyoxide.org/eb16b6ab4ab7ba61f33e2dfd0051e9a589dab601`

As well as the published key on (search for the fingerprint `0xEB16B6AB4AB7BA61F33E2DFD0051E9A589DA` - `https://pgp.mit.edu` - `https://keys.openpgp.org` - `https://keyserver.ubuntu.com`

You should then import it manually by issuing the following command on any OS:

```
gpg --import AnonymousPlanet_0x89DAB601_public.asc
```

Finally, verify the asc signature file (links above) against the PDF files by issuing the following commands:

```
gpg --verify guide.pdf.asc guide.pdf" gpg --verify guide-dark.pdf.asc guide-dark.pdf"
```

This should output a result showing it matches and it's ok.

How to verify the the authenticity and integrity of the files using Minisign:

To verify the files with Minisign:

- You should first dowbload minisign from https://jedisct1.github.io/minisign/
- Download the files along with their *.minisig signature file (they should be in the same directory)
- Download the Minisign public key available on the website and repository: `minisign.pub` (again place it in the same directory for convenience)
- Run the following command in a command prompt or terminal: `minisign -Vm guide.pdf -p minisign.pub`
- Output should show `Signature and comment signature verified`

How to check the safety of the files using VirusTotal:

The PDF and ODT files in this guide have been checked by VirusTotal, see the links below but do not trust them blindly and check the hashes matches and re-upload to VT if needed (**Note that this guide does not endorse VirusTotal. It should be used with extreme caution and never with any sensitive files due to their privacy policies**): - Light Theme: [VirusTotal] - Dark Theme: [VirusTotal] - ODT file: [VirusTotal]

Additional manual safety checks for the PDF files:

For additional safety; you can always double check the PDF files using PDFID which you can download at `https://blog.didierstevens.com/programs/pdf-tools/` (You might be wondering why should trust a random python script? Well it's open-source and well-known. It's probably a safer bet than trusting a random PDF).

Here are the steps:

- Install latest 3.9.x version of Python on your OS, Download PDFID and, from a command prompt or terminal, run:

```
python pdfid.py file-to-check.pdf
```

And you should see the following entries at 0 for safety, this 0 means there is no Javascript or any action that could possibly embed malicious scripts. Normally this won't be neceessary as most modern PDF readers won't execute those scripts anyway.

```
/JS                     0 #This indicates the presence of Javascript
which could be malicious
/JavaScript             0 #This indicates the presence of Javascript
which could be malicious
/AA                     0 #This indicates the presence of automatic action
on opening
/OpenAction             0 #This indicates the presence of automatic action
on opening
/AcroForm               0 #This indicates the presence of AcroForm which
could contain malicious JavaScript
/JBIG2Decode            0 #This indicates the PDF uses JBIG2 compression
which could be used for obfuscating malicious content
/RichMedia              0 #This indicates the presence rich media within
the PDF such as Flash
/Launch                 0 #This counts the launch actions
/EmbeddedFile           0 #This indicates there are embedded files within
the PDF
/XFA                    0 #This indicates the presence of XML Forms within
the PDF
```