How to Run Tor Arti as an HTTP Proxy or Behind a g3 HTTP Proxy

2025-09-01

How to run Tor Arti as an HTTP proxy or behind a g3 HTTP proxy.

Introduction

This is a short guide on how to use Tor through an HTTP proxy.

The setup uses three components:

- arti
- g3proxy
- Any software that supports an HTTP proxy

The diagram below shows the overall flow. {% diagram() %}

flowchart LR A[Client] –>|HTTP-protocol| B[g3] –>|SOCKS-protocol| Carti –> D[TOR-Network] –> G[Original Destination] E[Client] –>|HTTP-protocol| B[g3] F[Client] –>|HTTP-protocol| Carti

D[TOR-Network] –> H[Original Destination] D[TOR-Network] –> I[Original Destination]

J[Client] –>|SOCKS-protocol| Carti {% end %}

Arti

Quote from the Arti homepage:

> Arti is a complete rewrite of the C Tor codebase, and it is currently under active development. It is written in Rust, and it is designed to be modular, reusable, and easy to audit.

You need to build Arti before you can run it :smile:.
I am using an Ubuntu-based distribution, so this guide uses `apt`.

This is a short summary of the original Compiling Arti guide.

```
# clone the repo
git clone https://gitlab.torproject.org/tpo/core/arti.git

# navigate to the directory
cd arti

## install some necessary packages
```

```
sudo apt install libsqlite3-dev

cargo build -p arti --release
```

tip

You may need to run the build/install process more than once while adding missing dependencies on your build host.

g3proxy

You can either download prebuilt g3proxy packages or build it yourself.
I chose to build it.

```
git clone https://github.com/bytedance/g3
cd g3

sudo apt install liblua5.4-dev capnproto libc-ares-dev

cargo build
```

tip

You may need to run the build/install process more than once while adding missing dependencies on your build host.

g3proxy has many more configuration options than shown here. See the project README.

Config

g3proxy requires a configuration file.

```
---
runtime:
  thread_number: 2

log: stdout

server:
  - name: http_proxy
    escaper: default
    type: http_proxy
    listen:
      address: "[::]:10087" # <<<< http-proxy listening port
    tls_client: { }   # Open layer-7 https forward forwarding support

escaper:
```

```
  - name: default
    type: proxy_socks5
    proxy_addr: "[::1]:9150" # <<<< arti listening port
```

Runtime

After both tools (Arti and g3proxy) are built, open two shells: run Arti in one shell and g3proxy in the other.

Shell 1: Arti

```
/datadisk/git-repos/arti $
# target/release/arti proxy
2025-09-01T15:11:17Z  INFO arti::subcommands::proxy: Starting Arti 1.5.0
in SOCKS proxy mode on localhost port 9150 ...
2025-09-01T15:11:17Z  INFO tor_memquota::mtracker: Memory quota tracking
initialised max=8.00 GiB low_water=6.00 GiB
2025-09-01T15:11:17Z  INFO arti_client::client: Using keystore from "/
home/alex/.local/share/arti/keystore"
2025-09-01T15:11:17Z  INFO tor_dirmgr: Marked consensus usable.
2025-09-01T15:11:17Z  INFO tor_dirmgr: Loaded a good directory from
cache.
2025-09-01T15:11:17Z  INFO arti::subcommands::proxy: Sufficiently
bootstrapped; system SOCKS now functional.
2025-09-01T15:11:17Z  INFO arti::socks: Listening on [::1]:9150.
2025-09-01T15:11:17Z  INFO arti::socks: Listening on 127.0.0.1:9150.
2025-09-01T15:11:17Z  INFO tor_dirmgr: Marked consensus usable.
2025-09-01T15:11:17Z  INFO tor_dirmgr: Directory is complete. attempt=1
2025-09-01T15:11:17Z  INFO tor_guardmgr::guard: We have found that guard
[scrubbed] is usable.
2025-09-01T15:11:18Z  INFO arti::reload_cfg: Successfully reloaded
configuration.
2025-09-01T15:11:28Z  INFO tor_guardmgr::guard: We have found that guard
[scrubbed] is usable.
```

Shell 2: g3proxy

```
/datadisk/git-repos/g3 $
# target/debug/g3proxy --config-file g3proxy/examples/chain_socks_proxy/
g3proxy.yaml
```

Now configure your browser to use the HTTP proxy at `127.0.0.1:10087`, then test with `curl`.

{% diagram() %}

flowchart LR A[`curl`] –>|`127.0.0.1:10087`| B[g3] –>|`SOCKS`| Carti –> D[TOR-Network] –>|`https`| G[https://check.torproject.org/] {% end %}

Shell 3

```
# curl -sSL \
    --proxy 127.0.0.1:10087 \
    https://check.torproject.org/ \
    |egrep -i cong

      Congratulations. This browser is configured to use Tor.
      Congratulations. This browser is configured to use Tor.
```

Some Thoughts

Because `curl` also supports SOCKS, you can point `curl` directly to Arti.
However, many applications support only HTTP proxying and not SOCKS. For those applications, this setup is useful.
There are also other tools for HTTP(S)-to-SOCKS translation, for example Privoxy `forward-socks5`. I chose g3 because it is relatively small and written in Rust :smile:.

Update

Since Arti 1.7.0 (October 30, 2025), `HTTP CONNECT` is supported.
That means Arti can be used as an HTTP proxy endpoint, not only as a SOCKS5 endpoint.