Snowflake, a censorship circumvention system using temporary WebRTC proxies

Cecylia Bocovich
Arlo Breault
David Fifield
Serene
Xiaokang Wang

Authors are listed in alphabetical order.

USENIX Security 2024
2024-08-15

Paper home page: https://www.bamsoftware.com/papers/snowflake/
This talk: https://www.bamsoftware.com/talks/snowflake-usenix2024/

Précis

Snowflake is a censorship circumvention system—a way of enabling communication between network endpoints despite the interference of an intermediary censor. (Censors may do things like block IP addresses, send forged TCP RST packets, or falsify DNS responses.)

Snowflake uses a large pool of ultra-lightweight, temporary proxies ("snowflakes") that communicate using WebRTC protocols.

How does Snowflake resist address-based blocking?
: Its pool of temporary proxies is large (on the order of 100 K), and varies over time.

How does Snowflake resist content-based blocking?
: Transporting traffic in an encrypted WebRTC container.

Snowflake has been in serious deployment for 3+ years. It is a built-in circumvention option in Tor Browser, and serves a few tens of thousands of users at any time.

Number of users currently connected: 1

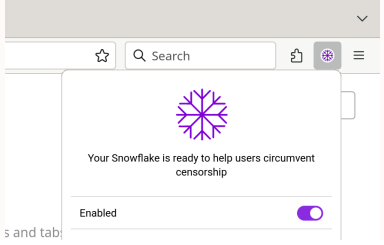
Number of users your Snowflake has helped circumvent censorship in the last 24 hours: 1

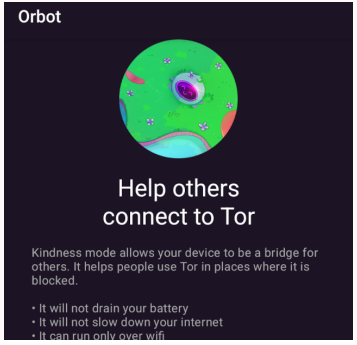
Enabled

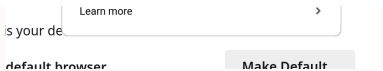
Learn more

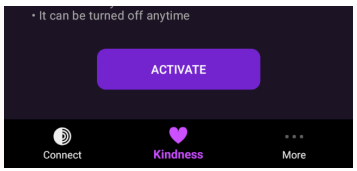
https://snowflake.torproject.org/

▼ or WebExtension proxy, Orbot kindness mode

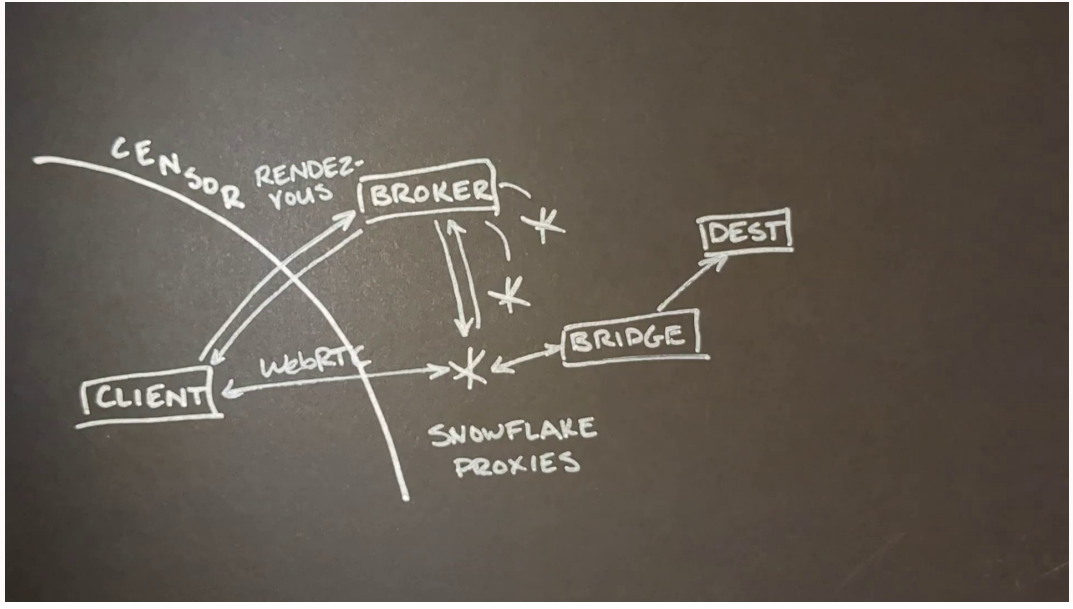




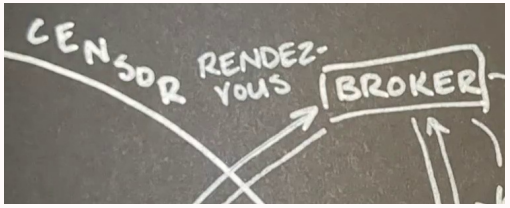

Learn more
s your de
default browser
Make Default


• It can be turned off anytime
ACTIVATE
Connect
Kindness
More

Snowflake system components



Rendezvous



The client sends its request for service through a secure *rendezvous* channel. Rendezvous is modular, and independent of the main WebRTC-based system.

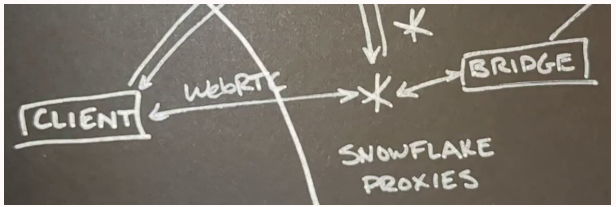
Currently deployed rendezvous methods:

- Domain fronting
- AMP cache
- Amazon SQS (Pu et al., FOCI 2024)

See "Communication Breakdown: Modularizing Application Tunneling for Signaling Around Censorship" (PETS 2024) for the rendezvous problem in general.

Session persistence

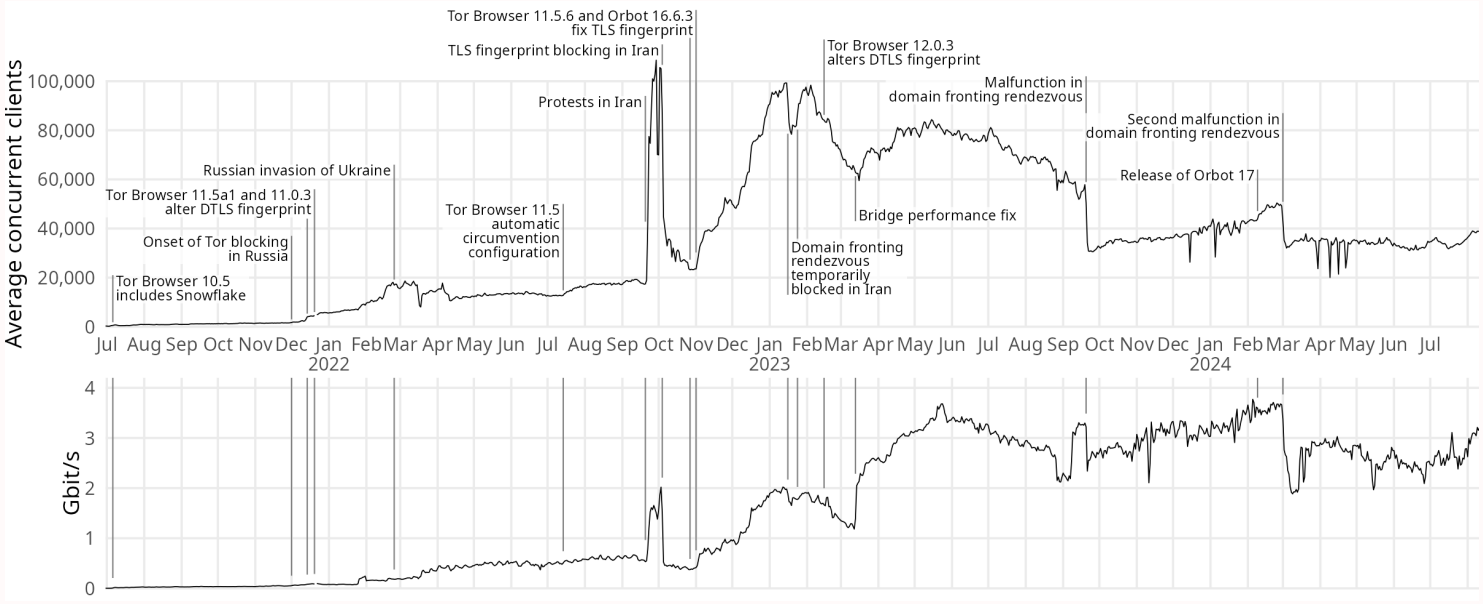
When an in-use proxy goes away, the client does another rendezvous and resumes the session on a new proxy.



This process uses end-to-end session state stored at the *client* and the *bridge* (a Turbo Tunnel design). Temporary Snowflake proxies are just pipes.

See "SpotProxy: Rediscovering the Cloud for Censorship Circumvention" (USENIX Security 2024) for an *active migration* that avoids the need for a repeated rendezvous.

Users and bandwidth



Snowflake users (daily average concurrent) and bandwidth (daily average).

More information in the paper

- Protocol fingerprinting
- NAT compatibility testing
- Proxy pool measurements
- Scalability and engineering challenges
- Experience reports against censorship in Russia, Iran, China, and Turkmenistan

---

Home page with documentation & source code
https://snowflake.torproject.org/
Paper home page
https://www.bamsoftware.com/papers/snowflake/
This talk
https://www.bamsoftware.com/talks/snowflake-usenix2024/
Donations for bridge hosting
"Snowflake Daily Operations" on OpenCollective

David Fifield <david@bamsoftware.com>