Turbo Tunnel, a good way to design censorship circumvention protocols

or, why your circumvention system should include an inner session layer

David Fifield
2020-08-11

https://www.bamsoftware.com/papers/turbotunnel/
2013

OSS – carries data in HTTP URLs

What’s in a URL

http://relay.com/91a37a20/6fe7703b/123/50/?return=192.0.2.5&data=UGxIYXlIG5vdGUgdGhh...

- Random (per-request) nonce to inhibit caching.
- Random (per-stream) ID to distinguish simultaneous clients.
- SEQ number.
- ACK number.
- Stream metadata: return address, FIN marker, suggested redirect method.
- Data payload.
2012

Code Talker Tunnel – uses UDP

Figure 4: SkypeMorph UDP packet body layout, where the size (in bytes) is under the name for each field. The shaded parts are encrypted using 256-bit AES counter mode. All bytes after the mac field are included in the HMAC-SHA256-64 computation.

https://censorbib.nymity.ch/#Moghaddam2012a Figure 4

StegoTorus – chops data and uses multiple cover channels

https://censorbib.nymity.ch/#Weinberg2012a Section 3
2013

meek – transfers data as a sequence of HTTP requests

```
POST / HTTP/1.1
Host: meek-reflect.appspot.com
X-Session-Id: cbIzfhx1Hn+
```

https://trac.torproject.org/projects/tor/wiki/doc/AChildsGardenOfPluggableTransports#meektransportlayer

“I lost time in the premature optimization of meek’s network performance. I was thinking about the request–response nature of HTTP, and how requests and responses could conceivably arrive out of order …. I made several attempts at a TCP-like reliability and sequencing layer, none of which were satisfactory. … I started implementing a version that strictly serialized HTTP requests and responses.”

ca. 2018

Snowflake – uses temporary browser-based proxies

When a proxy disappears, you want to be able to switch to a new one.

An inner TCP-like session and reliability protocol would make it easy...

Multiplex - one client splits traffic across multiple proxies
https://gitlab.torproject.org/tpo/anti-censorship/pluggable-transports/snowflake/-/issues/25723

New design for client -- server protocol for Snowflake
https://gitlab.torproject.org/tpo/anti-censorship/pluggable-transports/snowflake/-/issues/29206
Turbo Tunnel is my name for the design pattern of embedding a session/reliability layer in the middle of a circumvention protocol stack.

The session/reliability layer converts _streams_ to _packets_ and handles the details of retransmission and reassembly.

The obfuscation layer's only responsibility is to deliver encapsulated packets in a way that will not get blocked.
No need to invent a new session/reliability protocol:

- kcp-go
  https://github.com/xtaci/kcp-go
- quic-go
  https://github.com/lucas-clemente/quic-go

Prototyped Turbo Tunnel in several circumvention systems:

- obfs4
- meek
- Snowflake
- a DNS over HTTPS tunnel
If you want to help, you can try implementing a Turbo Tunnel design. There is example code and an example of converting a client–server program at
https://www.bamsoftware.com/papers/turbotunnel/example/

https://github.com/net4people/bbs/issues/9
david@bamsoftware.com