



Obfuscated VPN

LEAP's experience with Pluggable Transports
in the context of a VPN application

LEAP Encryption Access Project



LEAP VPN is a white label VPN designed for ease of use and developed for utility within censored environments. We work with trusted service providers to build and brand their VPN service. All aspects of LEAP's VPN, the server side and the application, are 100% open source.

LEAP VPN is the shared codebase for Bitmask, CalyxVPN, RiseupVPN, and SurVPN

Recent Developments at LEAP



User Centered Design Methodology

Work with Simply Secure to center rapid prototyping and small batch user testing

Pluggable Transports

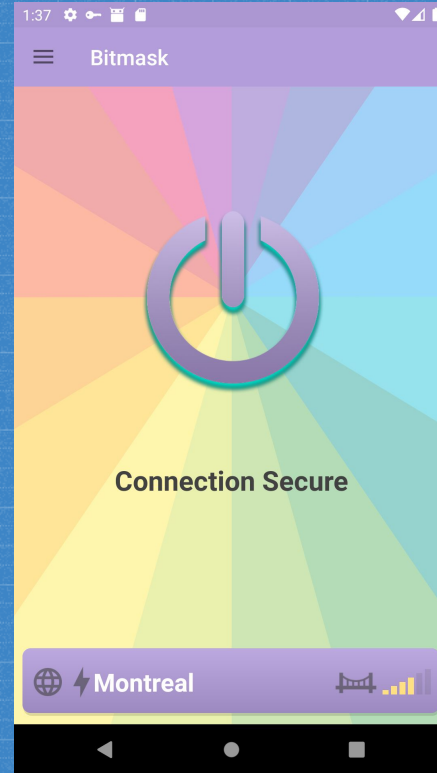
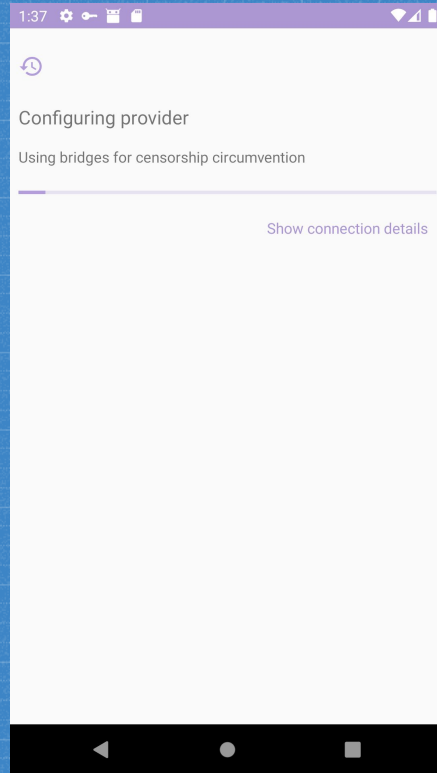
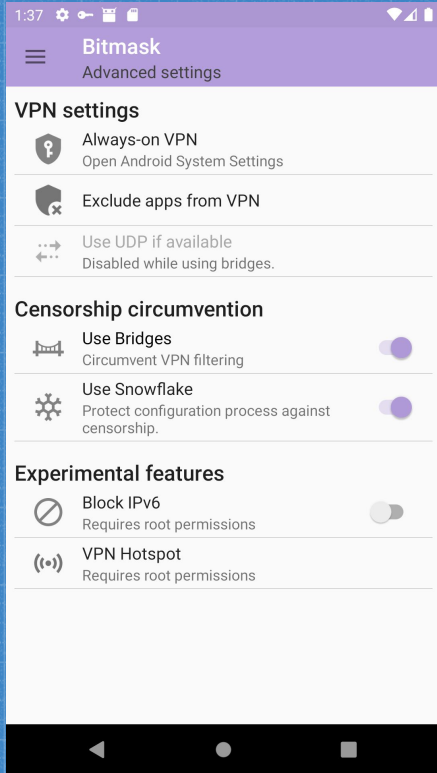
With funding from Internews, we worked with our partner providers, The Calyx Institute and Riseup to provide obsf4 and Snowflake PTs.

Bitmask

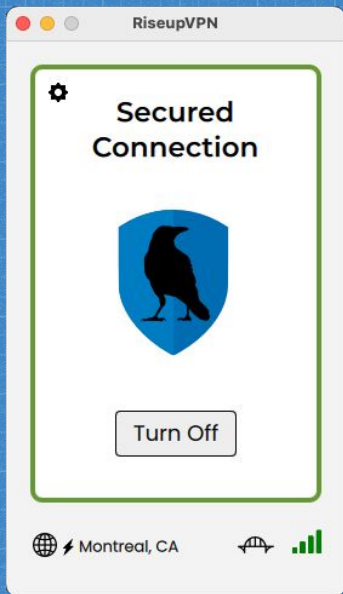


LEAPS multi-provider Android app. Users can choose either Riseup or Calyx as the provider.

Bitmask and PTs



RiseupVPN & PTs



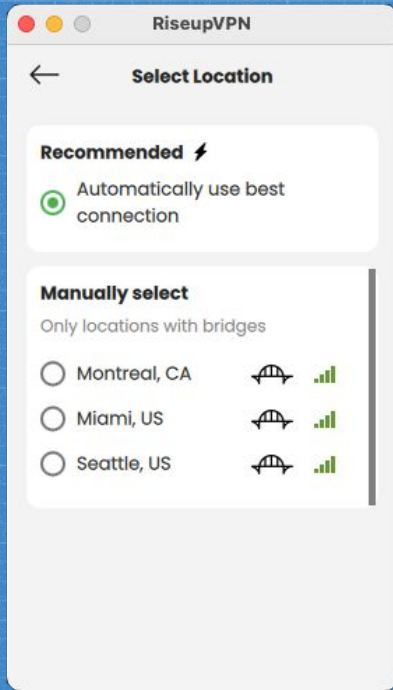
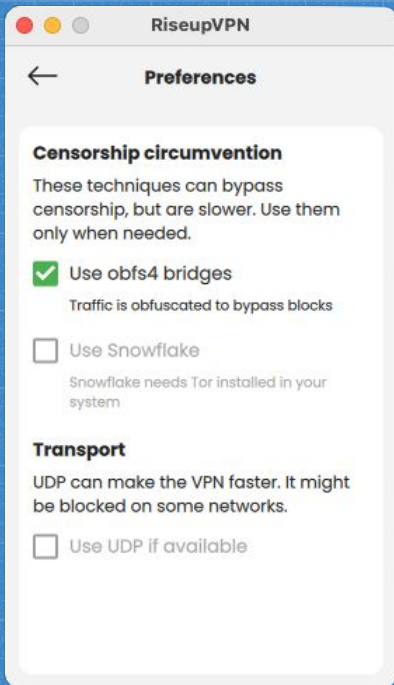
Pluggable Transports

RiseupVPN allows users to use obsf4 and snowflake on three of their six gateway locations.

Current events

In the last three weeks the number of daily users of LEAP VPN has increased by 25%, driven mostly by users in Russia. (Side note: This is overloading their gateways and they are looking for rapid response funding)

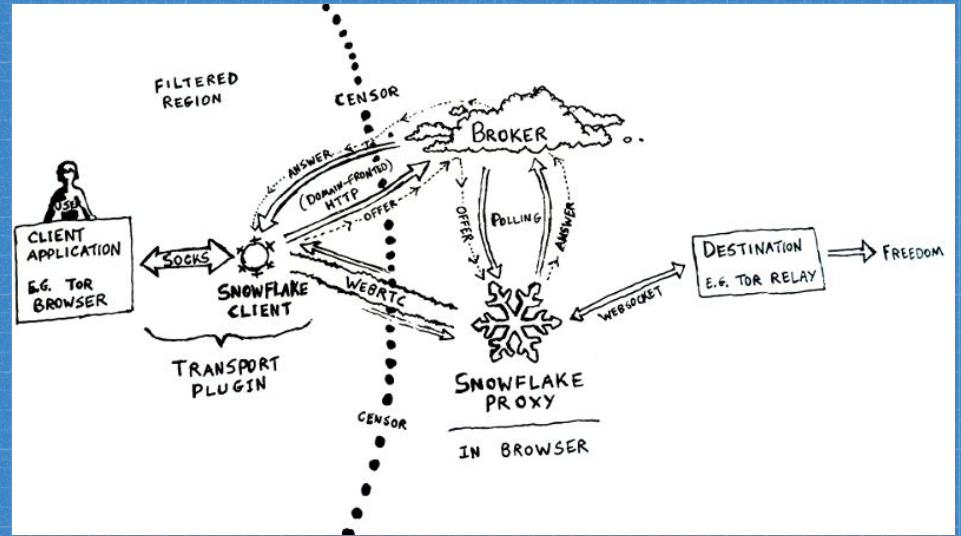
RiseupVPN + PTs



Pluggable Transports

In 2018, prompted by complaints from RiseupVPN users regarding blocked VPN, we started investigating Pluggable Transports.

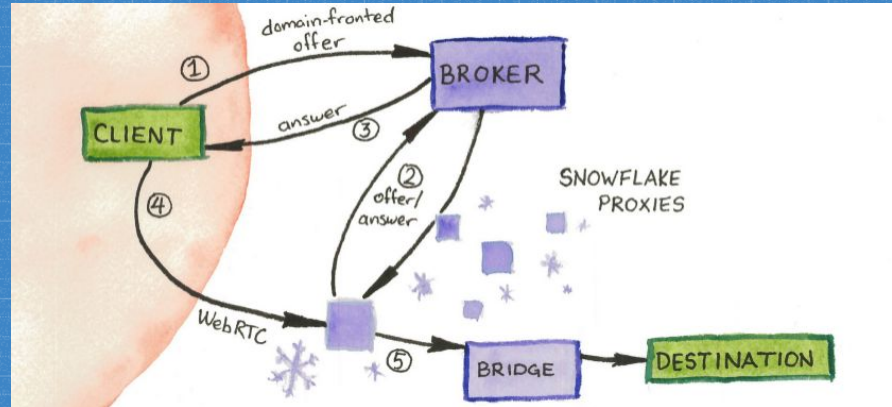
In 2019, in partnership with Calyx we received our first grant from Internews to work on PTs.



Pluggable Transports

Since then we have focused on developing and providing

- 1) **obsf4**
- 2) **obsf4 over UDP**
- 3) **Snowflake**





1

obfs4

obsf4 Implementation

We made it easy to deploy obs4 endpoints on our containerized platform

We moved the old puppet-based infrastructure to a more modern orchestration framework with containers

We implemented a Shapeshifter lib and heavily relied on Shapeshifter Dispatcher for cross-platform compatibility and seamless integration into our clients

obsf4 Implementation

UX Considerations

We didn't want the obsf4 endpoint to be overwhelmed with regular users, how to achieve that? Make clear that is only to be used in case of necessity, and also that it will likely slow your connection *.

We decided to emulate Tor terminology:

Let's call it "bridges"



* Not necessarily true, though.

obsf4 in the field

Problem

Implementing obfs4 based obfuscation helped to circumvent the VPN traffic to some extent

However some users were unable to use the VPN because the configuration process had been easily blocked by DNS.

obsf4 in the field

Solution

High impact and low effort was to implement an IP-mapping fallback mechanism for blocked DNS in the beginning of 2020

It allowed users to pass the GFW for a while, tested with users

obsf4 in the field

Lessons Learned

Consider what else do we need to take into account besides PTs for effective censorship circumvention?

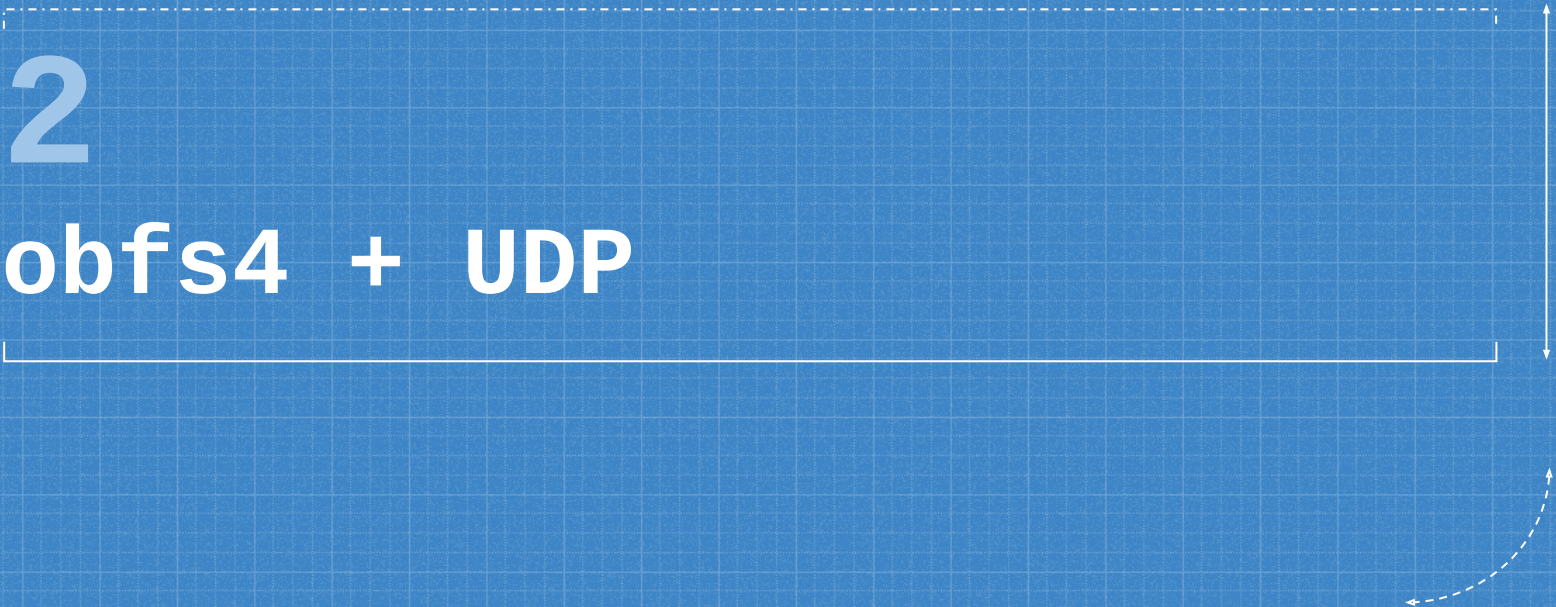
“Dumb” solutions might still bring much benefit at least in some censorship cases / regions:

Strategy: raise operational costs for censors.

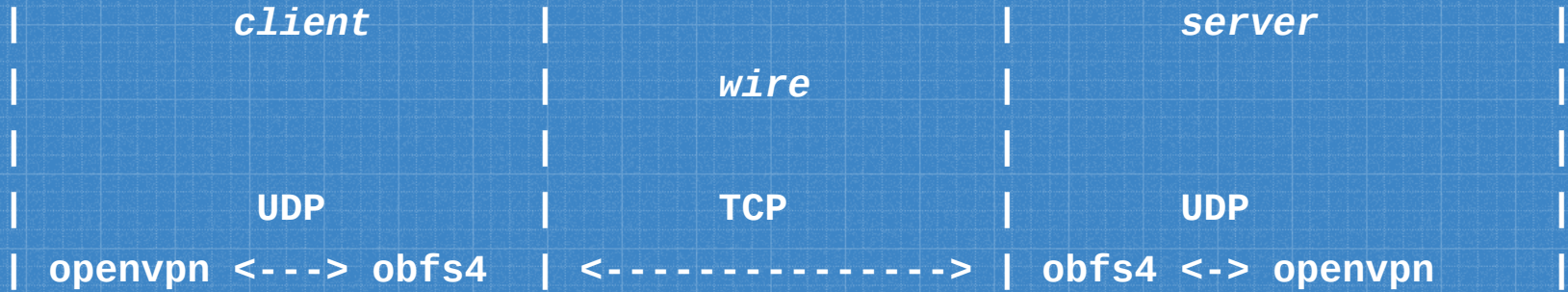


2

obfs4 + UDP



UDP mode and PTs



... not what the user expects (or needs)



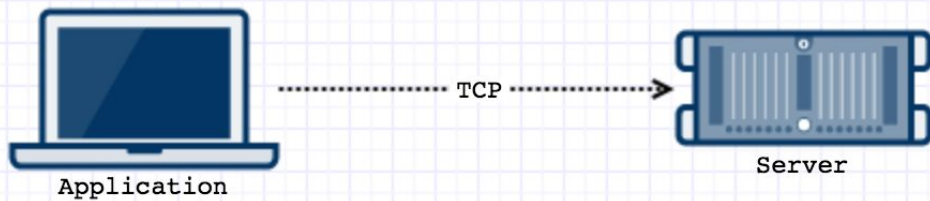
obsf4 and UDP

Why? faster, and for what we know UDP is less blocked in certain places.

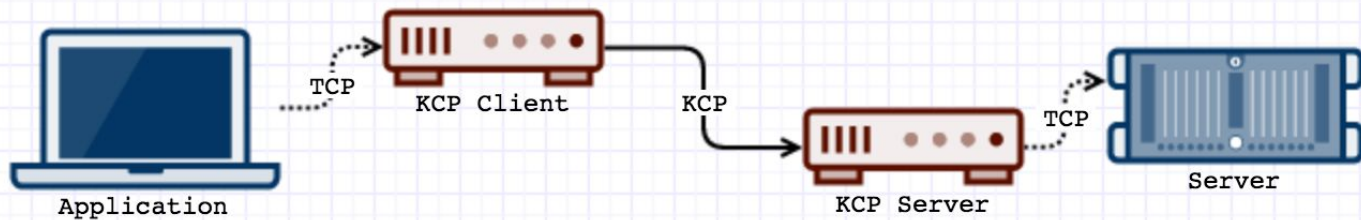
There is a fundamental complexity with using UDP and obsf4. UDP-OpenVPN would be encapsulated inside a TCP protocol, so it would be wrong to call it a UDP "transport" (and doing application layer UDP proxying on top of a obsf4 TCP transport does not provide so much benefit for anti-censorship on the OpenVPN case).

To address this we will produce a PT that encapsulates obsf4 within KCP.

- Pros: reliability over UDP.
- Cons: easy to fingerprint.



OPTIMIZE CONNECTIONS WITH KCPTUN





3

Snowflake



Snowflake



Can we leverage Snowflake for Censorship Circumvention? Two strategies explored:

- 1) **Bootstrap a parallel infra:**
too complex for now (many moving parts).
- 2) **VPN bootstrap only**

Snowflake

VPN bootstrap only

Tor as a dependency (*Potentially as a library.*)

“Last resort”: when contacting API is not possible + configurable option.

Pros:

- Great traction, many volunteers.
- Mature codebase, different proxy implementations.

Cons:

- slow bootstrap – need to inform user of progress.
- No guarantee the VPN connection is going to succeed.

Snowflake

VPN bootstrap only

The Future:

- Can we tunnel all traffic? (bootstrap time + bandwidth).
 - Match expectations + threat model.
- Realization: we need to ground research into empiricism (Tschantz et al, 2016).
 - Instrument production code:
 - successful sessions
 - failed attempts, etc.
 - Collaborate with OONI.

Lessons learned



Lessons Learned

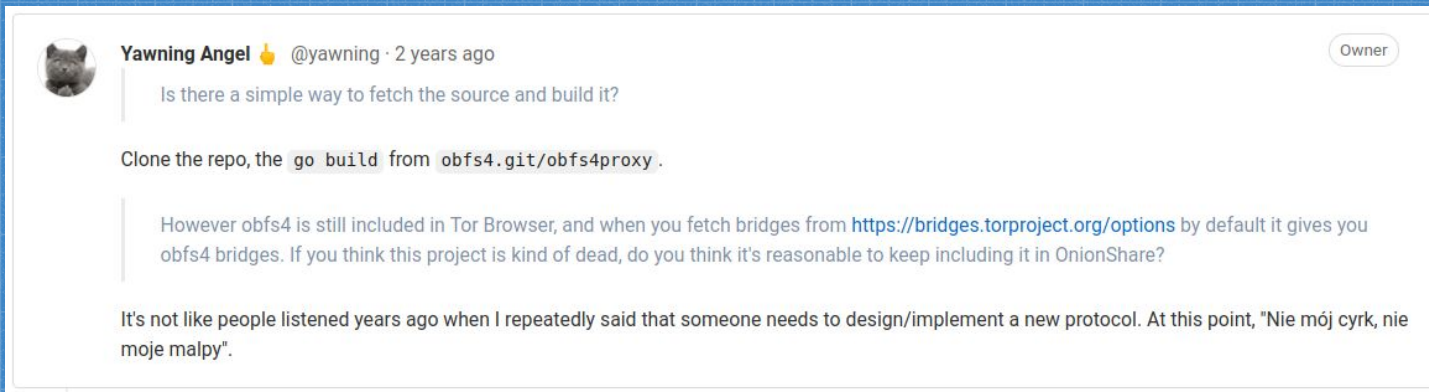


You don't want to burn that bridge (yet).

- Lower the cost to redeploy bridges + proxies to bridges (current work).
- Private bridges (current work).

Lessons Learned

Upstream code (and protocols) needs maintenance, its shelf life expires. Need to give love to track bugs and work with upstream (“we” kept using something because it “just works”: wrong move!)



The screenshot shows a GitHub issue comment. At the top left is a small circular profile picture of a cat. To its right, the text reads "Yawning Angel 🙄 @yawning · 2 years ago". In the top right corner of the comment box, there is a pill-shaped button labeled "Owner". The main text of the comment asks, "Is there a simple way to fetch the source and build it?". Below this, there is a code block containing the text: "Clone the repo, the `go build` from `obfs4.git/obfs4proxy`.". A horizontal line separates this from the next paragraph, which says: "However obfs4 is still included in Tor Browser, and when you fetch bridges from <https://bridges.torproject.org/options> by default it gives you obfs4 bridges. If you think this project is kind of dead, do you think it's reasonable to keep including it in OnionShare?". The final paragraph reads: "It's not like people listened years ago when I repeatedly said that someone needs to design/implement a new protocol. At this point, "Nie mój cyrk, nie moje malpy"."

Even with flaws, it's surprising that this **look-like-nothing protocol** has been working well (Adoption threshold? Lazy censors?).

Lessons Learned

Monitoring censorship of the own infrastructure becomes difficult if you don't want to hand over all your bridges information to the censor.

(At the same time, if you only measure "burned" bridges, you will be measuring the past).

Questions?



Twitter: @leapcode



Irc: #leap @libera



SlidesCarnival icons are editable shapes.

This means that you can:

- Resize them without losing quality.
- Change fill color and opacity.

Isn't that nice? :)

Examples:

