Characterizing and bypassing censorship remotely with the Outline SDK, remote proxies and AI

Splintercon - June 13, 2024

Vinicius Fortuna, Jigsaw (**fortuna@google.com**)

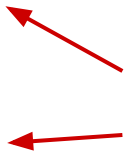
JIGSAW

Outline SDK

Reusable, composable and cross-platform components to empower your app against censorship

Outline SDK

- Libraries - add network resilience to your app
  - Transports - shadowsocks, tls, websocket
  - Proxy protocols - shadowsocks, socks5, http
  - Proxyless strategies - encrypted DNS, packet manipulation, happy eyeballs, strategy finding
  - VPN - ”tun2socks”
  - Mobileproxy - integrate into mobile apps
- Command-line Tools - experiment, measure, prototype
  - Fetch
  - Resolve
  - Local proxy



We’ll focus on these

Tools

JIGSAW

Tools - Resolve DNS

https://www.kaggle.com/code/vinifortuna/characterizing-and-bypassing-censorship-remotely-w/edit

```
go run github.com/Jigsaw-Code/outline-sdk/x/examples/resolve@33018ef
-resolver 8.8.8.8 -type A 1.2.3.4.nip.io
1.2.3.4

go run github.com/Jigsaw-Code/outline-sdk/x/examples/resolve@33018ef
-resolver 8.8.8.8 -type CNAME www.youtube.com
youtube-ui.l.google.com.
```

Tools - Fetch Page

```
go run github.com/Jigsaw-Code/outline-sdk/x/examples/fetch@33018ef
https://checker.soax.com/api/ipinfo | jq 'del(.data.ip)'
{
  "status": true,
  "reason": "",
  "data": {
    "carrier": "",
    "city": "North Bergen",
    "country_code": "US",
    "country_name": "United States",
    "isp": "Digital Ocean",
    "region": "New Jersey"
  }
}
```

Tools - Local Proxy

```
go run github.com/Jigsaw-Code/outline-sdk/x/examples/http2transport@33018ef -localAddr
127.0.0.1:8080 -transport
"ss://Y2hhY2hhMjAtaWV0Zi1wb2x5MTMwNTpHUDVTN1E3NlV5RFBTVElYUmRwM1d0@206.189.133.20:57510/"
Proxy listening on 127.0.0.1:8080

curl -s -p -x http://127.0.0.1:8080 https://checker.soax.com/api/ipinfo | jq 'del(.data.ip)'

{
  "status": true,
  "reason": "",
  "data": {
    "carrier": "",
    "city": "Bengaluru",
    "country_code": "IN",
    "country_name": "India",
    "isp": "Digital Ocean",
    "region": "Karnataka"
  }
}
```

Specifying Strategies

Specifying Strategies

https://pkg.go.dev/github.com/Jigsaw-Code/outline-sdk/x/config

Proxy Protocols

Shadowsocks proxy (compatible with Outline's access keys, package github.com/Jigsaw-Code/outline-sdk/transport/shadowsocks)

```
ss://[USERINFO]@[HOST]:[PORT]?prefix=[PREFIX]
```

SOCKS5 proxy (currently streams only, package github.com/Jigsaw-Code/outline-sdk/transport/socks5)

```
socks5://[USERINFO]@[HOST]:[PORT]
```

USERINFO field is optional and only required if username and password authentication is used. It is in the format of username:password.

Specifying Strategies

Transports

TLS transport (currently streams only, package github.com/Jigsaw-Code/outline-sdk/transport/tls)

The sni parameter defines the name to be sent in the TLS SNI. It can be empty. The certname parameter defines what name to validate against the server certificate.

```
tls:sni=[SNI]&certname=[CERT_NAME]
```

WebSockets

```
ws:tcp_path=[PATH]&udp_path=[PATH]
```

Specifying Strategies

DNS Protection

DNS resolution (streams only, package github.com/Jigsaw-Code/outline-sdk/dns)

It takes a host:port address. If the port is missing, it will use 53. The resulting dialer will use the input dialer with Happy Eyeballs to connect to the destination.

```
do53:address=[ADDRESS]
```

DNS-over-HTTPS resolution (streams only, package github.com/Jigsaw-Code/outline-sdk/dns)

It takes a host name and a host:port address. The name will be used in the SNI and Host header, while the address is used to connect to the DoH server. The address is optional, and will default to "[NAME]:443". The resulting dialer will use the input dialer with Happy Eyeballs to connect to the destination.

```
doh:name=[NAME]&address=[ADDRESS]
```

Address override.

This dialer configuration is helpful for testing and development or if you need to fix the domain resolution. The host parameter, if not empty, specifies the host to dial instead of the original host. The port parameter, if not empty, specifies the port to dial instead of the original port.

```
override:host=[HOST]&port=[PORT]
```

Specifying Strategies

Packet manipulation

These strategies manipulate packets to bypass SNI-based blocking.

Stream split transport (streams only, package github.com/Jigsaw-Code/outline-sdk/transport/split)

It takes the length of the prefix. The stream will be split when PREFIX_LENGTH bytes are first written.

```
split:[PREFIX_LENGTH]
```

TLS fragmentation (streams only, package github.com/Jigsaw-Code/outline-sdk/transport/tlsfrag).

The Client Hello record payload will be split into two fragments of size LENGTH and len(payload)-LENGTH if LENGTH>0. If LENGTH<0, the two fragments will be of size len(payload)-LENGTH and LENGTH respectively. For more details, refer to github.com/Jigsaw-Code/outline-sdk/transport/tlsfrag.

```
tlsfrag:[LENGTH]
```

Specifying Strategies

Composable

```
doh:name=cloudflare-dns.com&address=cloudflare.net:443 | tlsfrag:1
```

DNS-over-HTTPS

“Address fronting”

TLS Record Fragmentation

Specifying Strategies

SOCKS5-over-TLS, with domain-fronting:

```
tls:sni=decoy.example.com&certname=[HOST] | socks5:[HOST]:[PORT]
```

Onion Routing with Shadowsocks:

```
ss://[USERINFO1]@[HOST1]:[PORT1] | ss://[USERINFO2]@[HOST2]:[PORT2] |
ss://[USERINFO3]@[HOST3]:[PORT3]
```

Remote Access

JIGSAW

Options

Outline Server

```
-transport "ss://Y2hhY2hhMjAtaWV0Zi1wb2x5MTMwNTpHUDVTN1E3NlV5RFBTVElYUmRwM1d0@206.189.133.20:57510/"
```

SOCKS5 over SSH

```
ssh -D 127.0.0.1:1080 -C -N $USER@$HOST:$PORT
```

```
-transport "socks5://localhost:1080"
```

SOAX

```
-transport
socks5://package-${SOAX_PACKAGE}-sessionlength-3600-sessionid-${SESSION_ID}-country-ir-isp-mtn%20irancell:${SOAX_KEY}@proxy.soax.com:5000
```

SOAX

- Data extraction platform to extract data from the web.
  - People running nodes are paid and have plausible deniability
  - Measurement traffic is tiny compared to scraping
- Registered in the UK, CEO from Russia.
- Public Ethical Guidelines
- Hurdles:
  - can’t access every domain, but can access any IP
  - only ports 80, 443 and 5222
  - $99/month. Expensive, but allows a lot of testing, can be shared broadly.
  - requires face and id scan for identity verification
- Further investigation is welcome

Using SOAX

```
TRANSPORT="socks5://package-${SOAX_PACKAGE}-sessionlength-3600-sessionid-IRANCELL${SESSION_ID}-country-ir-isp-mtn%20irancell:${SOAX_KEY}@proxy.soax.com:5000"
go run github.com/Jigsaw-Code/outline-sdk/x/examples/fetch@33018ef -timeout 15 \
-transport
"socks5://package-${SOAX_PACKAGE}-sessionlength-3600-sessionid-IRANCELL${SESSION_ID}-country-ir-isp-mtn%20irancell:${SOAX_KEY}@proxy.soax.com:5000" \
https://checker.soax.com/api/ipinfo | jq 'del(.data.ip)'
```


```
{
  "status": true,
  "reason": "",
  "data": {
    "carrier": "MTN Irancell",
    "city": "Tehran",
    "country_code": "IR",
    "country_name": "Iran",
    "isp": "MTN Irancell",
    "region": "Tehran"
  }
}
```

Notebook

Try the tools yourself!

https://www.kaggle.com/code/vinifortuna/using-outline-sdk-tools/edit

Remote Measurements

JIGSAW

Detecting blocking of Youtube in Irancell

```
export
TRANSPORT="socks5://package-${SOAX_PACKAGE}-sessionlength-3600-sessionid-IRANCELL${SESSION_ID}-co
untry-ir-isp-mtn%20irancell:${SOAX_KEY}@proxy.soax.com:5000"

go run github.com/Jigsaw-Code/outline-sdk/x/examples/fetch@33018ef -timeout 15 -transport
"${TRANSPORT}|override:host=$(dig +short www.youtube.com | tail -1)" https://www.youtube.com/

2024/06/13 13:57:03 HTTP request failed: Get "https://www.youtube.com/": context deadline
exceeded (Client.Timeout exceeded while awaiting headers)

exit status 1
```

Bypassing blocking of Youtube in Irancell

```
go run github.com/Jigsaw-Code/outline-sdk/x/examples/fetch@33018ef -timeout 15 \
-transport "${TRANSPORT}|override:host=$(dig +short www.youtube.com | tail -1)|tlsfrag:1" \
https://www.youtube.com/ | grep -oe '<title>.*</title>'
```

```
<title>YouTube</title>
```

```
go run github.com/Jigsaw-Code/outline-sdk/x/examples/fetch@33018ef -timeout 15 -transport
"${TRANSPORT}|tlsfrag:1|doh:name=cloudflare-dns.com&address=www.cloudflare.net"
https://www.youtube.com/ | grep -oe '<title>.*</title>'
```

```
<title>YouTube</title>
```

Bypassing blocking of Youtube in Irancell

With Outline:

```
go run github.com/Jigsaw-Code/outline-sdk/x/examples/fetch@33018ef -timeout 15 \
-transport
"${TRANSPORT}|ss://Y2hhY2hhMjAtaWV0Zi1wb2x5MTMwNTpnVFVqWVhqdmoyc01HTU5zamNSd3ZE@206.189.133
.20:5222" \
https://www.youtube.com/ | grep -oe '<title>.*</title>'
```

```
<title>YouTube</title>
```

Data Analysis

JIGSAW

Let’s do some measurements!

Notebook:

https://www.kaggle.com/code/vinifortuna/measuring-website-censorship

Blocking of Outline in Russia

https://www.kaggle.com/code/vinifortuna/analysis-of-outline-blocking-in-russia-may-2024

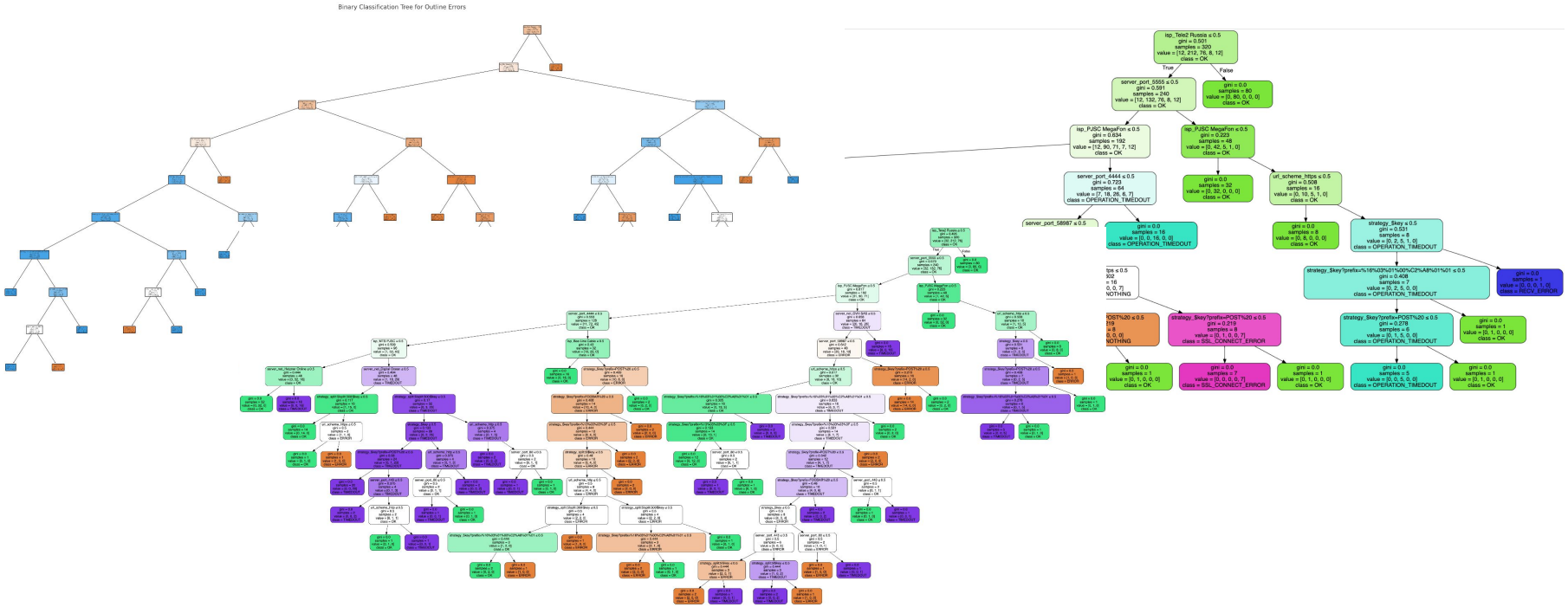
Takeaways:

- No blocking observed on Tele2.
- No blocking observed on Bee Line when using DigitalOcean. OVH can be used with the prefix POST%20.
- No blocking observed on MTS when using a high or ephemeral port, and blocking was independent of the server network.
- MegaFon was the most strict, but no blocking was observed when using DigitalOcean or Hetzner and the prefix POST%20.

Avoid targeted Clouds, use high port numbers and prefix POST%20

Using ChatGPT

https://chatgpt.com/c/267132a6-7118-4d4e-8b47-a9d595d4d13d

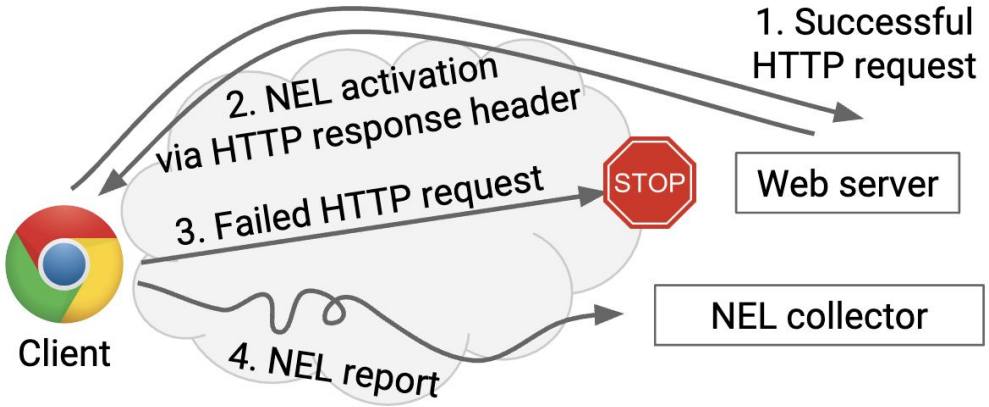


Network Error Logging

Network Error Logging

Issue: Usage metrics only capture successes, not failures

Solution: Client reports failures to an independent endpoint (different domain & IP)



Network Error Logging

Measuring Blocking of Fully Encrypted Protocols in Iran & Russia

https://lookerstudio.google.com/c/reporting/8d74d614-eabb-4878-a127-fa7d04fbbb36/page/tx40D

Discussion on Github

Contact Amir Gharabaghi, OTF IFCP fellow hosted by Jigsaw

Takeaways and opportunities

- Understand censorship
  - Do you own measurements and analysis with the Outline SDK
  - Leverage remote proxies. Deploy Outline Server, leverage SSH servers or use a network like SOAX
  - Measure both websites and circumvention strategies
  - Make Blockathon match real world behavior
- Find new strategies
  - Lots of variables, need to adapt strategies to context
- Test Block simulation against real world
- All materials and code on the conference folder
- Discuss on the Outline SDK Discussion Group