OpenVPN is Open to VPN Fingerprinting

USENIX Security 2022

Diwen Xue, Reethika Ramesh, Arham Jain, Michalis Kallitsis
J. Alex Halderman, Jedidiah R. Crandall, Roya Ensaﬁ

University of Michigan, Merit Network, Inc., Arizona State University
Internet traffic is increasingly being disrupted, tampered with, and monitored by ISPs, advertisers, and other threat actors.

VPN are on the Rise

"From 2010 to year-end 2019, the use of VPNs has increased by approximately four times" Cybersecurity Company PC Matic, 2020
From Enterprise Security
To
Privacy and Censorship
Circumvention

- Create private network across the public Internet through Encrypted Tunneling.
- Increasingly being used in non-enterprise setting.
An Evolving Threat Model

- Most of past research focused on the **Integrity** and **Confidentiality** of the tunnel.
  - Tunnel Penetrating Attacks
  - Data Injection
  - Traffic Leaks
- Threat actors now attacking **Availability**.

### Table III: Providers with traffic leakages—26 providers leak traffic during tunnel failure. * indicates those with traffic leaks
Not a hypothetical threat...

VPN Ban: Indian Parliamentary Committee Wants To Ban VPN Services In India
Virtual Private Network services or VPN could be in danger in India as the Parliamentary Standing Committee On Home Affairs is looking to...

Rain throttles Internet speeds for customers on VPNs
Jamie McKane 1 February 2021

Russia adds another VPN to its ban list
Last year, Russia banned Hola!VPN, ExpressVPN, KeepSolid VPN Unlimited, Nord VPN, Speedify VPN, and IPVanish VPN.
“Obfuscated” VPN services

Stealth VPN - the best solution to bypass restrictions in China

Stealth VPN works where ordinary VPN does not

Use obfuscated servers for extra privacy

- Hide your VPN use
- Avoid government censorship
- Bypass restrictions at work

Get Started

How the IPVanish Scramble feature works

IPVanish offers an obfuscation setting for OpenVPN on Windows, macOS, Android, and Fire TV devices called Scramble. This feature works by encoding and shuffling OpenVPN data packets so that tools meant to block VPN traffic let it pass.
Can ISPs and governments identify VPN traffic in near real time?

Can they do so at-scale, without incurring significant collateral damage from false positives?
We focus on OpenVPN and its variants!

The most popular protocol for commercial VPN services

“Obfuscated” VPN services built on top of OpenVPN

Mechanisms in place to impede fingerprinting attempts
Is OpenVPN Open to fingerprinting, *in practice*?

- Previous work used machine learning models on flow-level statistics
  - Connection duration
  - Inter-packet latency
  - Traffic symmetry

- Do these approaches work in practice?
  - Real-world ML-based censorship system not documented
  - Synthetic dataset, lab-based evaluations
  - Seemingly low false-positive can still be economically impractical
Effective investigation of Fingerprintability requires not only to identify vulnerabilities, but also to **demonstrate practical exploits** under the constraints of **how ISPs and censors operate** in the real world.
Deployment inside Merit Network
Deployment inside Merit Network

1. Client Station initiates VPN connections

2. All traffic is being mirrored to Monitoring Station (Filter)

Passing Traffic

Internets

Merit Network

0.0.0.1:1194

0.0.0.2:1194
Deployment inside Merit Network

1. Client Station initiates VPN connections
2. All traffic is being mirrored to Monitoring Station (Filter)
3. Filtering results are forwarded to probing system
4. Probing targets are sent to dedicated Probers
5. Probers send probes to confirm detection results
6. Confirmed VPN endpoints are stored in a database for manual analysis
Deployment inside Merit Network

Examining how the Great Firewall of China discovers hidden circumvention servers. IMC’15
Analyzing China’s blocking of unpublished Tor bridges. FOCI’18
How China detects and blocks Shadowsocks. IMC’20
Fingerprinting OpenVPN

Filtering Phase:
- Opcode Evolution (Byte Pattern)
- ACK Repetition (Packet Size)

Probing Phase:
- Customized Probes
  (Server Behaviors)
Fingerprint 1: Opcode

- Opcode is a fixed value in the header which denotes each stage of the session.
- Opcode evolution of a new OpenVPN session is unique and can be used to fingerprint OpenVPN.
- Flexible enough to catch certain “obfuscated” variants.

**Opcode message types:**

```c
#define P_CONTROL_HARD_RESET_CLIENT_V1 1
#define P_CONTROL_HARD_RESET_SERVER_V1 2
#define P_CONTROL_SOFT_RESET_V1 3
#define P_CONTROL_V1 4
#define P_ACK_V1 5
#define P_DATA_V1 6
#define P_DATA_V2 9
#define P_CONTROL_HARD_RESET_CLIENT_V2 7
#define P_CONTROL_HARD_RESET_SERVER_V2 8
#define P_CONTROL_HARD_RESET_CLIENT_V3 10
```
Fingerprint 2: ACK Packets

- Explicit acknowledgement and retransmission model for “control” messages.
- Uniform in size for each session; not the same as TCP ACK flag;
- Quantify “ACK Fingerprint” as a set of threshold-based detection rules.
Detection accuracy of Filtering phase

Filtering Phase:

- Opcode Evolution (Byte Pattern)
- ACK Repetition (Packet Size)

- All Flows: 3.3 Billion
- Persistent Flows: 1.9 Million
- Filter Outputs: 15835
- Prober Outputs: 519
Detection accuracy of Filtering phase

Filtering Phase:

- Opcode Evolution (Byte Pattern)
- ACK Repetition (Packet Size)

Increasing accuracy to prevent significant collateral damage requires active probing.

- All Flows: 3.3 Billion
- Persistent Flows: 1.9 Million
- Filter Outputs: 15835
- Prober Outputs: 519
Active Probing

- Defense mechanisms “tls-auth” and “tls-crypt” enable a firewall-like protection.
  - OpenVPN remains silent until the client proves knowledge of a shared secret.
- Application may stay silent, but application-specific behaviors can still be observed at network level.

(related: Detecting Probe-resistant Proxies NDSS'20)

<table>
<thead>
<tr>
<th>ProbeName</th>
<th>Probe Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>BaseProbe 1</td>
<td>x00x0ex38.{8}x00x00x00x00x00</td>
</tr>
<tr>
<td>BaseProbe 2</td>
<td>x00x0ex38.{8}x00x00x00x00</td>
</tr>
<tr>
<td>TCP Generic</td>
<td>x0dx0ax0dx0a</td>
</tr>
<tr>
<td>One Zero</td>
<td>x00</td>
</tr>
<tr>
<td>Two Zero</td>
<td>x00x00</td>
</tr>
<tr>
<td>Epmd</td>
<td>x00x01x6e</td>
</tr>
<tr>
<td>SSH</td>
<td>SSH-2.0-OpenSSH_8.1\r\n</td>
</tr>
<tr>
<td>HTTP-GET</td>
<td>GET/HTTP/1.0 /\r /n /\r /n</td>
</tr>
<tr>
<td>TLS</td>
<td>Typical Client Hello by Chromium</td>
</tr>
<tr>
<td>2K-Random</td>
<td>Random 2000 Bytes</td>
</tr>
<tr>
<td>ProbeName</td>
<td>Probe Content</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------------------------------------------</td>
</tr>
<tr>
<td>BaseProbe 1</td>
<td>x00x0ex38.{8}x00x00x00x00x00</td>
</tr>
<tr>
<td>BaseProbe 2</td>
<td>x00x0ex38.{8}x00x00x00x00</td>
</tr>
<tr>
<td>TCP Generic</td>
<td>x0dx0ax0dx0a</td>
</tr>
<tr>
<td>One Zero</td>
<td>x00</td>
</tr>
<tr>
<td>Two Zero</td>
<td>x00x00</td>
</tr>
<tr>
<td>Epmd</td>
<td>x00x01x6e</td>
</tr>
<tr>
<td>SSH</td>
<td>SSH-2.0-OpenSSH_8.1/r/n</td>
</tr>
<tr>
<td>HTTP-GET</td>
<td>GET/HTTP/1.0 /r /n /r /n</td>
</tr>
<tr>
<td>TLS</td>
<td>Typical Client Hello by Chromium</td>
</tr>
<tr>
<td>2K-Random</td>
<td>Random 2000 Bytes</td>
</tr>
</tbody>
</table>
Testing on Commercial VPNs

- Effective in detecting vanilla OpenVPN flows. (39/40 vanilla configurations)

- Surprisingly, 72.67% obfuscated flows also detected. (34/41 obfuscated configurations).
  - “Obfuscated” VPN services use OpenVPN as backbone protocol
  - Insufficient obfuscation failing to mask fingerprints.
Fingerprinting “Obfuscated” VPNs

- XOR Obfuscation
- Additional Encrypted Tunneling
- Obfuscated Servers
## Fingerprinting “Obfuscated” VPNs

<table>
<thead>
<tr>
<th>XOR Obfuscation</th>
<th>Additional Encrypted Tunneling</th>
<th>Obfuscated Servers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:1 correspondence between opcodes and ciphertext</td>
<td>Lack of random padding</td>
<td>Co-location of Bridges and vanilla servers.</td>
</tr>
</tbody>
</table>
XOR Obfuscation

- Unofficial patch that scrambles payloads by a series of XOR operations.
- Opcode excluded from reversal, therefore always mapped to the same ciphertext. Behavior preserved in multiple implementations.
Accuracy

- Collateral damage as the fundamental measure of practicality.
  - Week-long evaluation, aggregated 20 Gbps of mirrored traffic.
  - 3,638 flows flagged. (0.0039%)
  - Manual analysis found supporting evidence for 90% of flagged connections.
Conclusion

- Fingerprinting OpenVPN is within the reach of any network operator.
  - Even with obfuscation patches deployed in the wild.
  - Risk of throttling, blocking, and even follow-up attacks on VPN tunnel.
  - Users should *NOT* expect unobservability, even with “stealth” VPN.
Conclusion

- Fingerprinting OpenVPN is within the reach of any network operator.
  - Even with obfuscation patches deployed in the wild.
  - Risk of throttling, blocking, and even follow-up attacks on VPN tunnel.
  - Users should *NOT* expect unobservability, even with “stealth” VPN.

- Moving forward...
  - Short-term defense.
  - A gap between obfuscation research and implementation.
OpenVPN is Open to VPN Fingerprinting

USENIX Security 2022

VPNalyzer.org

Diwen Xue, Reethika Ramesh, Arham Jain, Michalis Kallitsis
J. Alex Halderman, Jedidiah R. Crandall, Roya Ensafi

University of Michigan, Merit Network, Inc., Arizona State University
Backup
Encrypted Tunneling

- Tunnel-based obfuscation wraps OpenVPN traffic through encryption.
  - SSL/SSH Tunnel, obfs234 ...
- ACK fingerprints are still observable outside **tunnels that lack random padding**.

Obfuscation Servers

- In practice, most of obfuscation servers — “Bridges” — are co-located with vanilla TCP servers. (34/41 for /29 subnet)
- Infrastructures are shared between obfuscated and vanilla services from different providers.
**Probe 1 & Probe 2**

<table>
<thead>
<tr>
<th>ProbeName</th>
<th>Probe Content</th>
<th>Expected Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>BaseProbe 1</td>
<td>x00x0ex38.8</td>
<td>Explicit ServerReset or Short Close</td>
</tr>
<tr>
<td>BaseProbe 2</td>
<td>x00x0ex38.8</td>
<td>Long Close</td>
</tr>
<tr>
<td>TCP Generic</td>
<td>x0dx0ax0dx0a</td>
<td>Short Close</td>
</tr>
<tr>
<td>One Zero</td>
<td>x00</td>
<td>Long Close</td>
</tr>
<tr>
<td>Two Zero</td>
<td>x00x00</td>
<td>Short Close</td>
</tr>
<tr>
<td>Empl</td>
<td>x00x01x6e</td>
<td>Short Close</td>
</tr>
<tr>
<td>SSH</td>
<td>SSH-2.0-OpenSSH_8.1/1</td>
<td>Short Close</td>
</tr>
<tr>
<td>HTTP-GET</td>
<td>GET/HTTP/1.0/1/n/1/n</td>
<td>Short Close</td>
</tr>
<tr>
<td>TLS</td>
<td>Typical Client Hello by Chromium</td>
<td>Short Close</td>
</tr>
<tr>
<td>2K-Random</td>
<td>Random 2000 Bytes</td>
<td>Short Close &amp; RST</td>
</tr>
</tbody>
</table>
Is OpenVPN Open to fingerprinting, in practice?

- **Real-world ML-based censorship system not documented**
- **Synthetic dataset, lab-based evaluation**
  - Same dataset [ISCXVPN2016][3,14,15,17,24,26,68]
- **Seemingly low false positive rate can be misleading.**
  - (1% FPR, 0.01% Base Rate 1 in 100 blocked is actually VPN)