VPNalyzer: Systematic Investigation of the VPN Ecosystem

Reethika Ramesh
University of Michigan
reethika@umich.edu

Leonid Evdokimov
Independent
leon@darkk.net.ru

Diwen Xue
University of Michigan
diwenx@umich.edu

Roya Ensafi
University of Michigan
ensafi@umich.edu

***Abstract*—Use of Virtual Private Networks (VPNs) has been growing rapidly due to increased public awareness of online risks to privacy and security. This growth has fueled the VPN ecosystem to expand into a multi-billion dollar industry that sees a frequent influx of new VPN providers. Nevertheless, the VPN ecosystem remains severely understudied, and the limited research concerning VPNs has relied on laborious manual processes. There is a need for a solution which empowers researchers and average users to investigate their VPN providers.**

In this work, we present *VPNalyzer*, a system that enables systematic, semi-automated investigation into the VPN ecosystem. We develop a cross-platform tool with a comprehensive measurement test suite containing 15 measurements that test for aspects of service, security and privacy essentials, misconfigurations, and leakages. Using the *VPNalyzer* tool, we conduct the largest investigation into 80 desktop VPNs.

Our investigation reveals several previously unreported findings highlighting key issues and implementation shortcomings in the VPN ecosystem. We find evidence of traffic leaks during tunnel failure in 26 VPN providers, which seriously risk exposing sensitive user data. We are the first to measure and detect DNS leaks during tunnel failure, which we observe in eight providers. Overall, we find a majority of providers lack IPv6 support, and five even leak IPv6 traffic to the user's ISP. We observe that adoption of practices we consider security and privacy essentials is not uniform across VPN providers. Multiple providers share underlying infrastructure, and 29 providers use third-party, public DNS services. Alarmingly, 10 VPN providers leak traffic even in their most secure configuration, with six leaking data even with a "kill switch" feature enabled. Our results highlight the effectiveness of *VPNalyzer* in finding issues even in the most popular VPN providers. Consumer Reports used *VPNalyzer* in their efforts to create data-driven recommendations for their users.

I. Introduction

Internet service providers, advertisers, and online threat actors are increasingly disrupting, tampering with, and monitoring Internet traffic [17], [48], [78], [81]. High-profile security incidents, widespread reports of ISPs selling data about their users, and the increasing prevalence of geographic discrimination have fueled an increased public awareness of online risks and access restrictions [7], [54], [76]. As a result, the use of virtual private networks (VPNs) has been growing rapidly, not only among activists and journalists but also among average users [4], [6], [30], [51]. This trend has been further accelerated by more people working from home due to the COVID-19 pandemic [31]. Notably, statistics from Egypt, France, the UK, and the US point to a surge in VPN adoption over the past year [53].

Despite being a growing multi-billion dollar [61] industry, the VPN ecosystem remains severely understudied. Previous security evaluations of VPN products [26], [38], [43] have been limited in the scale and types of VPN products analyzed and have used inconsistent heuristics that prevent monitoring of issues in the VPN ecosystem over time. Specifically, the latest reliable investigation into the VPN ecosystem was performed in 2018, as a one-time study of mostly free and trial versions of commercial products [43]. These previous studies, though valuable, all involved a large amount of manual effort.

The VPN ecosystem is extremely dynamic, with constant changes in the features offered with new providers frequently entering the market. This means that a large-scale and continuous empirical assessment of the VPN ecosystem requires methodology that can scale easily across many providers and can be repeated across time, thus making manual investigation as in previous work impractical. Any solution should ultimately empower researchers and average users with an extensible and convenient tool that facilitates investigation into VPN providers.

In this work, we present ***VPNalyzer***—a system that enables systematic, semi-automated investigation into the VPN ecosystem—and perform large-scale empirical assessments of 80 popular VPN providers using *VPNalyzer*. As part of the *VPNalyzer* system, we build a cross-platform tool that has a comprehensive measurement test suite combined with a simple installation and user interface. *VPNalyzer* is also designed to be modular and configurable to facilitate additions and upgrades to adapt to frequent changes of the VPN ecosystem. Our tool is equipped with 15 measurements that test for aspects of service, security and privacy essentials, misconfigurations, and leakages including whether the VPN has implemented an effective mechanism to protect users during tunnel failure. All in all, we cover essential tests from previous work, with six measurements that take direct inspirations, and nine new measurements for which we implement our own methods.

Using the *VPNalyzer* tool, we conduct the largest state-of-the-art investigation into desktop VPNs on both MacOS and Windows, which includes free and paid VPN providers, as well as self-hosted VPN solutions, and our institutional VPN. In total,

Network and Distributed Systems Security (NDSS) Symposium 2022
24-28 April 2022, San Diego, CA, USA
ISBN 1-891562-74-6
https://dx.doi.org/10.14722/ndss.2022.24285
www.ndss-symposium.org

we have 230 experiments from 80 unique VPN providers. This study, in addition to contributing valuable insights about the providers, highlights the value and effectiveness of *VPNalyzer*.

Our investigation reveals several previously unreported findings highlighting key issues and implementation shortcomings in the VPN ecosystem. Surprisingly, we find that a majority of VPN providers do not support IPv6, and worse, five providers even leak IPv6 traffic to the user's ISP, including our own university VPN. We find evidence of traffic leaks during tunnel failure in 26 VPN providers which seriously risk exposing sensitive user data, especially to adversaries such as governments and ISPs that are capable of inducing such failures. More specifically, we are the first to measure and detect DNS leaks during tunnel failure, which we observe in 8 providers. Further, we observe that multiple VPN providers use the same underlying infrastructure, making colocated servers easier to block, and 29 providers (including paid ones) configure clients to use public DNS services. Two providers do not tunnel all user traffic in their default configuration, which deviates from users' expectation. Finally, we conduct a case study testing custom "secure" configurations of 39 top providers. Alarmingly, even in their secure configuration, 10 VPN providers leak traffic, six of which *even had a "kill switch" feature enabled*. These results are shocking considering that these VPNs are popular, with millions of users that trust them with sensitive data.

VPNalyzer is designed to empower researchers and users and to be easily adoptable as a user-friendly tool empowering the community to be vigilant about issues in the VPN ecosystem. *Consumer Reports*, a leading consumer research and advocacy organization, used *VPNalyzer* as part of their efforts to produce a data-driven and reliable recommendation for their millions of users [27]–[29]. Following our future public release, we hope that *VPNalyzer* benefits users and helps the general public choose better VPN providers.

II. Background and Related Work

The VPN ecosystem is especially dynamic due to the constant influx of new providers and frequent changes in their popularity. This has been attributed to a variety of factors, such as increasing user demand, varying censorship trends, prevalence of geographic restrictions on content, countries banning VPN use, providers' loss of reputation, and companies being acquired or rebranded [1], [4], [30], [42], [51], [53], [54].

Users of VPN products get conflicting advice from online recommendations and often lack the time and knowledge to conduct evaluations of their own. Moreover, VPN providers often employ marketing tactics that use jargon and exaggerated claims, making it hard for users to discern what is true. Different VPN providers also offer a variety of subscription models: free services, freemium models (sometimes with limited features), and paid VPN services that range anywhere from about \$5–\$20 a month, often with discounts for long-term plans. Although there has been a plethora of reports ranking these commercial VPNs, they are either limited, or lack objectivity and use inconsistent heuristics to evaluate VPN providers [22], [52]. There is a dearth of trusted, objective reviews and the few that exist are limited in scale, only capture a snapshot of the VPN ecosystem at the time, and are not repeated across time.

A small number of prior academic studies and closely related research efforts have analyzed VPN products. These studies, though limited, have been adept at identifying problems plaguing the ecosystem. In 2016, Ikram et al. performed static and dynamic analysis of 283 VPN permission–enabled Android apps and revealed serious privacy and security issues, including instances of malware in VPN apps' source codes [38]. In 2018, Khan et al. conducted an empirical analysis of 62 commercial VPN providers and found that many VPNs leak user traffic through a variety of means [43]. But they also found that commercial VPN providers are less likely to intercept or tamper with user traffic than previously studied forms of traffic proxying. However, they predominantly tested providers with *free and trial versions* of desktop applications or used OpenVPN configuration files. Another study explored vulnerabilities in 30 commercial VPN products focusing on the configuration of VPN clients and software [5] and found that vulnerabilities can stem from unsafe instructions to users, insecure third-party binaries, and use of fixed pre-shared keys. These studies, while valuable in measuring and identifying issues with VPN providers, involved a large amount of manual work and hence are not *scalable*, and *cannot be repeated easily*.

Other studies that focus on identifying vulnerabilities that exploit leakages and privilege escalation attacks demonstrate how adversaries can use these attacks to infer the identity of the user or execute arbitrary code. Perta et al. in their manual analysis of 14 popular VPN providers, identify developer-induced bugs and misconfigurations which lead to IPv6 and DNS leaks, which could deanonmyize users [70]. Fazal et al. showed how an attacker could penetrate into the VPN tunnel by exploiting VPN clients with a dual-NIC to bypass connection to the VPN server and gain control over the VPN tunnel [18].

Further, there have been studies focusing on verifying the locations of network proxies. VPN providers advertise servers in many countries with little proof of their claims. A study by Weinberg et al. [82] found that of the 2,269 servers studied, over one-third of them are definitely not in the geolocation or the country advertised, and another one-third might not be.

Finally, studies such as Netalyzr [46], IoT Inspector [33], Wehe [56] and others [14], [20], [49] paved the way for leveraging end-users to conduct network measurements, and they also provide insights to future measurement tools on effectively involving users in conducting end-host measurement [45].

The inconsistencies of online recommendations and the limitations of previous work have emphasized the need for a system that can that can help users perform systematic investigation of the VPN ecosystem. We fill this research gap with *VPNalyzer* and show its benefits and value by performing empirical assessments of 80 popular VPN providers.

III. *VPNalyzer* Design

Our aim is to build a system that has sufficient functionality combined with a simple installation process and user interface to empower average users to investigate different security and privacy aspects of VPN providers.

We build a cross-platform desktop application for Windows, MacOS, and Linux using the open-source `Electron` framework [16], chosen for its cross-platform compatibility and native

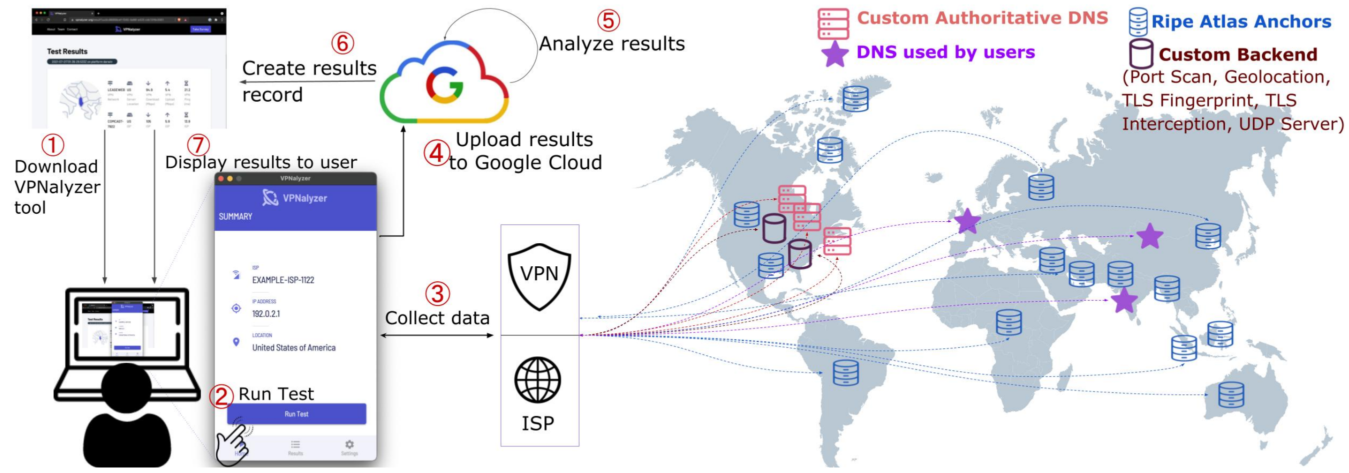
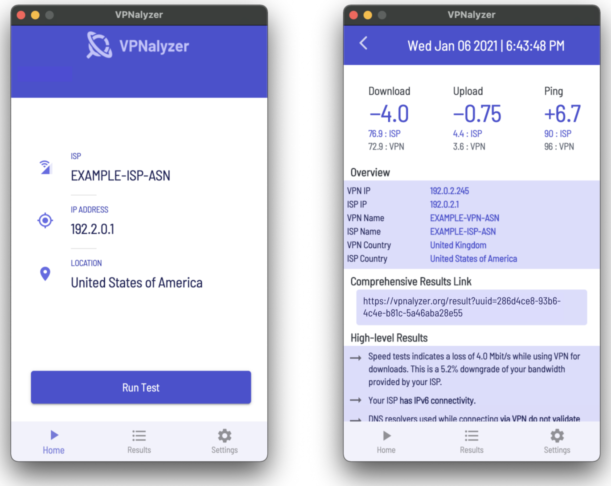


Figure 1: ***VPNalyzer* Architecture**— (1) User downloads application. (2) User installs the application, reviews our privacy policy, consents to be part of study. (3) User runs an "experiment" consisting of three stages: ensuring VPN is disabled and either granting or denying administrative privileges, enabling VPN and running *VPN case*, and disabling VPN and running *ISP case* (4) Once experiment is done, the application seeks explicit consent from user to upload experiment data to Google Cloud Storage. (5) Analysis pipeline works on the uploaded data. (6) Extracted results appear on website front-end. (7) User visits unique link pertaining to their "experiment" to view detailed results.◇

API availability. We develop the UI for the application with `React` and implement the measurements using `Node.js`. Initially, we explored creating a browser-based test suite, extension, or plugin. But none of these alternatives provide the level of functionality, fine-grained access for robust measurements, and convenience that a desktop application affords us. Furthermore, desktop VPNs have not been previously studied *at scale*, since methods to test them are notoriously hard to automate.

A. System Architecture and Components

Our system architecture (as described in Figure 1) starts with a user visiting our website to download the application for their specific platform in the form of a *.zip* file. Once the application is extracted and run, the user is first presented with our privacy policy and consent form (more details provided in §III-B). Upon agreeing, the user proceeds to the homepage where the user's current public IP address, Autonomous System (AS) Name, and geolocation are displayed, as shown in Figure 2.



ISP
EXAMPLE-ISP-ASN
IP ADDRESS
192.2.0.1
LOCATION
United States of America

Wed Jan 06 2021 | 6:43:48 PM

Download	Upload	Ping
−4.0	−0.75	+6.7
76.9 : ISP	4.4 : ISP	90 : ISP
72.9 : VPN	3.6 : VPN	96 : VPN

Overview

VPN IP	192.0.2.245
ISP IP	192.0.2.1
VPN Name	EXAMPLE-VPN-ASN
ISP Name	EXAMPLE-ISP-ASN
VPN Country	United Kingdom
ISP Country	United States of America

Comprehensive Results Link
https://vpnalyzer.org/result?uuid=286d4ce8-93b6-4c4e-b81c-5a46aba28e55

High-level Results
→ Speed tests indicates a loss of 4.0 Mbit/s while using VPN for downloads. This is a 5.2% downgrade of your bandwidth provided by your ISP.
→ Your ISP has IPv6 connectivity.
→ DNS resolvers used while connecting via VPN do not validate

Figure 2: ***VPNalyzer* Application**—The homepage (left) contains the user's current AS Name, public IP address, and geolocation detected using the public IP. The results page (right) contains a link to detailed results, and a summary displayed upon completion of an experiment.◇

Desktop Application: Currently, the *VPNalyzer* test suite contains 15 "measurements" that test for aspects of service, misconfigurations, leakages, and support for a set of security and privacy essentials, (more details in §IV). Each run of the application is termed an "experiment" that takes ≈20 minutes. An experiment flow is divided into three stages: bootstrapping in the ISP stage, performing measurements with VPN on (*VPN case*), and performing measurements with VPN off (*ISP case*). We perform the 15 measurements sequentially with the VPN and again without the VPN. This flow is necessary to confirm and corroborate our observations in the case of VPN leaks and misconfigurations. There are also essential background services, such as packet capture, that run throughout an experiment.

Building a system such as *VPNalyzer* comes with various technical challenges. Cross-platform development requires specialized knowledge especially since our tool requests administrative privileges. Designing a test suite conducive for testing VPNs, as well as ensuring that results are comparable between the VPN and ISP cases is a significant task. Further, considering the dynamic nature of the VPN ecosystem, *VPNalyzer* must be modular and configurable to facilitate additions and upgrades to the test suite. To facilitate broad distribution, our Windows and MacOS applications must be code-signed and notarized.

Experiment Flow: In the *bootstrapping in ISP stage*, the user is asked to confirm that the VPN is disabled. Then, the application runs bootstrapping code and prompts the user to optionally provide administrative privileges. If granted, packet captures are initialized which allows us to investigate any ambiguous results. If not, the application skips the privileged measurements and conducts all the others. In the next stage, *VPN case*, the user is asked to *turn on their VPN*, and upon confirming, the set of measurements for the VPN connection is performed. Then, in the *ISP case*, the user is asked to *turn off their VPN*, and the same measurements are repeated for the ISP connection.

Next, the user is prompted to answer a short survey about the VPN provider they tested. With their explicit consent, the packet captures and the experiment log are uploaded, and they may decline with minimal loss of client-side functionality. Finally, the user is presented with a preliminary results page and a link to more detailed findings, as shown in Figure 2.

Front-end and Backend Hosts: The *VPNalyzer* system uses several public and custom backend components necessary for different measurements as well as a website front-end which hosts the latest release of our application, results pages, and other miscellany about our research. We use reliable public services to serve as measurement helpers, for instance, the `RIPEStat Data API` to get public IP address and AS information and the Measurement Lab's `Locate Service (mlab-ns)` [55] to find the closest available M-Lab server.

Furthermore, we developed various custom backend components, such as authoritative DNS nameservers, port-scanner, custom UDP heartbeat server, TLS interception tester, and webserver hosting configuration files. We host backend components in academic institution subnets which are unlikely to be blocked, having multiple options for public DNS-over-HTTPS (DoH) servers, and hosting configuration files on a GitHub pages website. Considering that Google resources are unavailable in some countries, we instruct users to turn on their VPN and retry if the uploading of the experiment log fails. Additionally, to ensure availability, we implement a Prometheus monitoring system that alerts us in case any of our custom backends malfunction [72]. The complete list of public services and custom backends used for each measurement is in Table I.

Data Storage and Analysis: VPNalyzer uses `Google Cloud` [24] for both storage of the experiment data and our analysis pipeline that works on the data and creates the detailed results that will be shown on our website front-end to the user.

B. Ethics Considerations and Consent

Since our system involves collecting data by running measurements from users' machines, conducting ethical measurements and following good Internet citizenship are integral parts of our design principles. First we approached our institution's IRB, which determined our study to be "Not Regulated by the IRB," as we study the VPNs rather than the user/human subjects. Aiming to set a high standard for ethical measurement, we designed our measurements and system to follow the principles described in the Menlo report [15]. We highlight some of our key considerations below.

We offer users our detailed privacy policy and consent form before they can proceed to run any measurements. These documents were carefully designed following the language used by OONI's data policy document and inspired by the Harvard CyberLaw Clinic's guide to risks of security research [65], [69]. They were also shared with colleagues experienced in such studies, and revised using their feedback, ensuring that we adequately inform and help our users make an informed decision. We also provide means for users to contact us for more information.

We provide users with the final authority on the data our application collects. For example, the application requests administrative privileges, which are necessary to run certain measurements, but users are free to decline, in which case the application skips the privileged measurements and conducts the others. The application can collect packet captures during each experiment, but users are informed and must explicitly consent before packet captures are uploaded for each experiment.

We attempt to be good Internet citizens when conducting network measurements. When public services and endpoints are part of our measurements, we use them only for their intended purposes. Since our quality of service measurement depends on running the NDT7 (Network Diagnostic Test) using infrastructure operated by Measurement Lab (`M-Lab`) [55], we obtained their consent to use their servers. We contacted and obtained permission from npcap (the packet capture library for Windows) maintainers to bundle it with our application for Windows [63]. We also bootstrap most of the measurements in the ISP stage, due to the consideration that users may be paying for the VPN bandwidth.

Some countries have laws that prohibit using VPNs, so we explicitly inform users to be cognizant of these restrictions before using *VPNalyzer*. Since some tests are run without the VPN enabled, we inform users that their local ISP and VPN provider, and possibly their government, will be able to detect that the user is running *VPNalyzer*.

Finally, with respect to the issues we discovered, detailed in §VI, we have contacted the VPN providers and are in the process of disclosing our findings to each of them.

IV. *VPNalyzer* MEASUREMENTS TEST SUITE

In this section, we describe the 15 measurements that are performed during each experiment in the application. Every experiment is assigned a universally unique identifier (`UUID`), which is an RFC 4122 version 4 `UUID`, and the application creates an "experiment directory" named after the `UUID` under the application's data directory path. This directory contains the experiment log file and, if the user grants administrative privileges, the packet capture files.

Previous studies have identified key issues, known security and privacy flaws, leakages, and best practices for VPNs [13], [38], [43], [70], [82]. Our *VPNalyzer* test suite covers all the essential tests from previous work as well as implementing new measurements. While these tests are comprehensive and significantly improve the testing methods used in previous work, they do not necessarily cover *all* aspects of a VPN. For instance, we do not compare VPNs based on their usability or measure the logging policies of a VPN.

Our focus is on testing for a breadth of important issues and facilitating collection of extensive data about the ecosystem

Measurement Type	Measurement Name	Goal	Inspired by Previous Work	Public Services and Custom Backends Used
Aspects of Service	Bandwidth and Latency Tests	Calculate the performance penalty/overhead incurred by using the VPN	[47], [55]	`M-Lab` Infrastructure
	Geolocation Test	Fetch Cloudflare's IP geolocation and collect RTT measurements to `RIPE Atlas` Anchors	[9], [82]	Custom Cloudflare backend, `RIPE Atlas` Anchors, Anchors list Updater
	RPKI Validation	Test if the user's VPN and ISP are implementing BGP safely using RPKI validation		Cloudflare RPKI endpoints
Misconfigurations and Leakages	AS Mismatch	Detect possible IP leakage using AS Mismatches	[74]	`RIPEstat Data API`
	VPN Kill Switch Test	Detect whether VPN has implemented the kill switch feature correctly		`RIPEstat Data API`, custom UDP heartbeat servers
	DNS leak during tunnel failure	Detect whether VPN providers' killswitch mechanisms leak DNS traffic		Public DNS `whoami` helpers
Security and Privacy Essentials	Port Scan	Scan to discover different services running on the VPN server using `Nmap`		Custom port scanner backend
	Router Scan	Ascertain if the user's home (ISP) router's management interface is reachable while connected to the VPN		–
	DNS Discovery	Discover all possible DNS resolvers available to the user in both *VPN case* and *ISP case*		Public DNS `whoami` helpers, `RIPEstat Data API`
	Presence of DNS Proxy	Identify if the VPN has a DNS proxy that targets all the DNS resolvers used by the user's machine		Public recursive DNS resolvers, public DNS `whoami` helpers
	Support for DNSSEC	Test if an available resolver validates DNSSEC signatures		Custom domain, authoritative nameservers
	Use of QNAME Minimization	Test if an available resolver implements qmin	[13]	Custom domain, authoritative nameservers
	Lack of support for DoH and Presence of DNS64 Resolver	Test if an available resolver intentionally signals that the network is unsuitable for DoH, and if the resolver is a DNS64 resolver		Mozilla canary domain, `ipv4only.arpa`
	TLS Interception	Identify presence of TLS interception using the certificate	[41], [43], [78]	Certificate Fetching backend, Certificate Transparency logs
	TLS Fingerprinting	Identify presence of TLS interception using TLS fingerprint	[21]	TLS fingerprinting backend

Table I: **Measurements**—The goal of the 15 measurements performed by *VPNalyzer* and the custom backend and public services used for each measurement.◇

from the end user's machine. Going in-depth and investigating edge cases, while interesting, do not justify the additional complexity. For instance, our *support for DNSSEC* measurement checks if a resolver validates DNSSEC signatures but does not compare resolver behaviors with respect to different DNSSEC algorithms. Further, our measurement results are meant to be informative for the user, and for the tests that need further inspection such as the kill switch test and geolocation, we leverage our analysis pipeline to take extra steps such as verifying results using packet captures to prevent false positives before presenting them to the user. A short description of each of the measurements is given in Table I.

A. *Aspects of Service*

Bandwidth and Latency Tests: This measurement is designed to calculate the performance overhead incurred by using the VPN. We first find an `M-Lab` server closest to the VPN and use it to run several ndt7 (Network Diagnostic Tool) measurements over IPv4 and IPv6 in the *VPN case* and the *ISP case*. Finding an `M-Lab` server closest to the VPN is important in order to allow a fair comparison of the quality of service. If the server position is chosen based on the user's location, it may cause a bloat in the measured performance overhead due to the trombone effect [10].

We calculate the performance overhead by comparing the bandwidth and latency between the *VPN case* and the *ISP case*. We choose ndt7 over other public performance-measuring services such as iPerf3 or Ookla because we are not interested in measuring last-mile speed or approximating a browsing experience. Rather, we want to estimate the performance that is obtainable when the user is connected to a particular VPN server as compared to when they are connected to their ISP [47].

Geolocation Test: This measurement is designed to estimate the geolocation of the VPN server. Instead of using free IP-to-geolocation databases, which are notoriously unreliable and contain many errors especially regarding VPN services [23], [71], [82], we opt to conduct this measurement using the following two methods.

First, we leverage Cloudflare's IP geolocation service offered to site owners where the country code (in ISO 3166-1 Alpha 2 format) of the visitor's IP is included in the header of each request. To that end, we hosted a custom webserver at our University and added the domain to Cloudflare's free plan tier. The measurement makes a request to the domain and our webserver extracts the information from the `CF-IPCountry` request header [9]. Since this same information is typically used by Cloudflare to offer geo-specific services, we believe it is a good approximation to use for our purpose as well.

Our second approach uses the Weinberg et al. [82] upgraded Constraint-Based Geolocation algorithm (CBG++) by collecting round-trip time measurements to hosts in known locations. Similar to previous work, we use `RIPE Atlas` anchors as the "landmark" hosts since they are reliable, their documented locations are accurate, and conveniently, they continuously publish public databases containing the round-trip times between each other. Currently, there are over 723 `RIPE Atlas` anchors, and given their added reliability as compared to `RIPE Atlas` Probes, we choose to use only the anchors, which we fetch and update on our webserver daily.

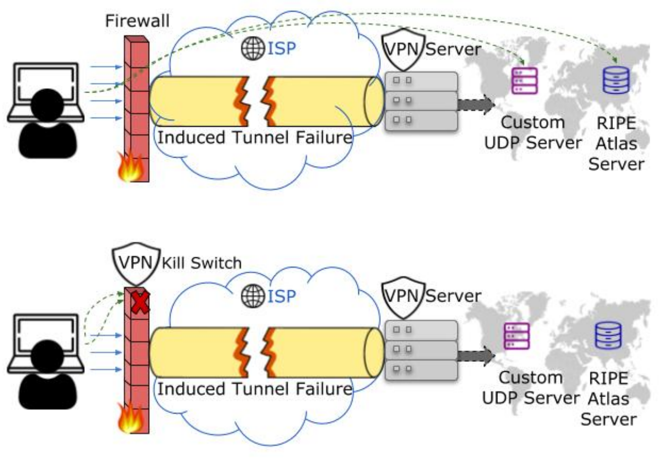


Figure 3: **VPN Kill Switch**— The top image depicts the scenario where the VPN kill switch feature is disabled, and the bottom image shows the kill switch feature enabled and working. Our measurement induces a tunnel failure by blocking all traffic not on our "allowlist". When the VPN kill switch is disabled (top), the traffic to the allowlist hosts is allowed to pass through the ISP link. In the bottom case, the traffic to the allowlist is also blocked due to the VPN's kill switch feature.⋄

Our measurement initiates a TCP connection to port 80 of each anchor, due to knowledge that a large number of VPN servers ignore ICMP ping requests and others drop TCP and UDP packets to unusual port numbers. Upon receiving a response from the anchor, it records the round-trip time for each query and saves it in the experiment log for future analysis.

The application displays the country name as reported from our Cloudflare custom webserver. Considering the amount of computation required for the Weinberg et al. method, we plan to publish them in an aggregate form because providing its result in real-time to the user is not feasible.

RPKI Validation: We conduct this measurement to test if the user's VPN provider (and ISP) implement the Border Gateway Protocol (BGP) safely. BGP is vulnerable to leaks and prefix hijacks, which can be mitigated by Resource Public Key Infrastructure (RPKI) [2]. The absence of RPKI validation or its incorrect implementation by a VPN provider network may allow an attacker to maliciously announce routes and disrupt traffic intended for the VPN server. To measure whether RPKI origin validation is properly implemented, we use Cloudflare's existing testing infrastructure used in `isbgpsafeyet.com`. Cloudflare provides two prefixes—one covered with a valid Route Origin Authorization (ROA) [3] and another with an invalid ROA. During this measurement, we make an HTTPS request to each of the sources. If the request to the invalid source fails while being able to fetch contents from the valid source, then it is an indication that RPKI validation is implemented in the network.

B. Misconfigurations and Leakages

AS Mismatch: This measurement is designed to detect possible IP leakage using Autonomous System (AS) information. For the user's public IPv4 and IPv6 addresses, we obtain the Autonomous System Numbers (ASNs) from `RIPEstat Data API's` "Network Info" data call [74]. In the *VPN case*, if the ASes of the IPv4 and IPv6 address (where available) do not belong to the same organization, it indicates that the traffic is being leaked to a "second" AS. If the AS(es) in the *VPN case* and *ISP case* are the same, it most likely indicates an IP packet leak, or it means that the VPN provider is using the same network as the ISP.

VPN Kill Switch Test: This measurement is designed to understand if the user's VPN has implemented an effective measure to protect users' traffic during tunnel failure. The kill switch feature, as it often called, is a fundamental security measure that is used to prevent any information leakage when VPN tunnel failure occurs. If the connection to the VPN server fails for any reason, this mechanism cuts off the user's connection to the Internet in order to disallow unprotected access until the VPN is able to reconnect. The functioning of the kill switch feature is illustrated in Figure 3.

Conceptually, to test for failures in such a mechanism, we create an "allowlist" of certain destination hosts, and then cause a tunnel failure by blocking all traffic *except to and from hosts on the allowlist*. If the VPN kill switch is effective, the traffic to the hosts on the allowlist should *also be blocked*. While this seems straightforward, it demands complex platform-specific implementation which we do in three stages: bootstrapping, setting up firewall rules to induce tunnel failure, and finally a two-pronged approach to detect VPN kill switch failures.

In the bootstrap stage of our application, we seek administrative privileges necessary for making changes to the user's firewall. We then initiate sessions with our two custom UDP servers to begin receiving UDP heartbeats, and log the firewall's state. In the *VPN case*, prior to running any measurements, we set up the necessary platform-specific components and log the firewall state again. This setup code on Linux entails creating new chains for the `iptables`. On Windows, we log the version of PowerShell and details about the `NetSecurity` module that is used to interact with the machine's firewall. On MacOS, we check if the `pf` program allows us to create custom anchors, verify that `pfctl` functions as expected, and finally enable the packet filter and obtain a token that is necessary to revert changes made by the measurement on the user's machine.

Next, in the *VPN case* we create a "allowlist" to be added to our custom firewall rules. This allowlist contains the IPv4 and IPv6 addresses of `RIPEstat Whats My IP`, **one** of our custom UDP heartbeat servers, and authoritative nameservers and public DNS resolvers belonging to Cloudflare, Google, and OpenDNS which become necessary for both our two-pronged approach, and the upcoming *DNS leak test*. We then modify the user's firewall, apply our custom rules to induce tunnel failure *without* resetting existing rules, and log the state of the firewall again. Once applied, the firewall blocks all traffic (to any port) to and from all hosts **not on the allowlist**, meaning there can be no communication to and from the VPN server.

Lastly, after inducing tunnel failure we use a two-pronged approach to detect flaws in the VPN kill switch:

- First, for over 120 seconds, we periodically query the IPv4 and IPv6 endpoints of `RIPEstat Who am I` data call that is on our allowlist. We chose 120 seconds because

OpenVPN has a timeout of 120 seconds to allow tunnel failure to be signalled [66] and the official OpenVPN Connect v3 client uses 60 seconds as the default connection timeout. If the kill switch feature is not enabled, the query will reach the endpoint and the IP returned will be the user's ISP IP address shown in Figure 3 (top). On the other hand, if the kill switch works effectively, the queries will time out, as shown in Figure 3 (bottom).

- Second, confirming induced tunnel failure requires another step of validation in case the VPN connection is governed by a stateful firewall, which typically preserves connection state once established. To that end, we use our custom UDP heartbeat servers to check if the firewall states are indeed preserved by a stateful firewall. Starting from the bootstrapping stage, our custom UDP servers, called Server_A and Server_B, continuously send UDP heartbeats to the application and receive acknowledgements. Next, in the allowlist, we only add Server_A and not Server_B. After the custom rules are applied, the heartbeat traffic from Server_B should be blocked. We can conclude that our custom rules have induced tunnel failure (i.e. no stateful firewall) when traffic from Server_A is also blocked.

In reporting the result of our *kill switch measurement*, we report leaks when the UDP heartbeat servers indicate successful tunnel failure and the user's ISP IP leak is confirmed from the `RIPEstat` endpoint's response.

Though the technique of using custom firewall rules to trigger tunnel failure has been used before, we identify a number of factors that can interfere with measurement results. Factors such as ordering of anchors in the main firewall, and VPNs inserting dynamic rules on the fly affect how the firewall rules are parsed, and hence may lead to false positives results if not handled carefully. Unfortunately, these factors were overlooked by previous studies. Specifically Khan et al. [43] triggered their blocking by resetting the test machine's firewall configuration with their own rules according to the code on their project's GitHub repository [44]. In doing so, they *possibly overrode* rules added by the VPN application, thereby effectively turning off many VPNs' kill switch mechanisms *before testing them*, leading to potential false positive results. Hence, we developed our own two-pronged approach to try and overcome such issues.

DNS Leak during tunnel failure: This measurement investigates if VPN providers' protection mechanisms during tunnel failure leak DNS traffic. Some VPN providers allow DNS queries to bypass their kill switch or firewall rules possibly to resolve domain names of their servers to attempt reconnection during tunnel failure. However, this behavior introduces privacy risks that can expose the user. For instance, during tunnel failure any third-party application on the device can send DNS queries to obtain the ISP IP of the user. Moreover, the ISP could learn the sites visited by the user via the 'A' and 'AAAA' queries made. These risks can be avoided by implementing more-secure reconnection methods. For instance, Wireguard resolves necessary DNS names during bootstrap [83].

To test for this DNS leak, we first resolve and add to our allowlist all the IPv4 and IPv6 IPs of the authoritative nameservers and recursive resolvers belonging to popular public DNS services such as Cloudflare, Google, and OpenDNS. Once the allowlist is applied as explained in the *VPN Kill Switch Test*, over the period of 120 seconds, we send two `whoami` DNS probes: one to a public recursive resolver and another to an authoritative nameserver, which are repeated every 100 milliseconds. Each probe is sent once over TCP and UDP for each round. These `whoami` probes are queries of the form: `dig +noedns -t txt -c chaos whoami.cloudflare. @one.one.one.one.` and `dig +noedns -t txt whoami.cloudflare.com. @ns3.cloudflare.com.` (similar queries to services of the other two providers) which return the public IP address in the response.

If no response is received for any of the queries, the test declares no DNS leak. Otherwise, we corroborate the received responses with collected packet captures to confirm that the user's ISP IP is returned, which we then consider a DNS leak.

C. Security and Privacy Essentials

The VPN ecosystem is largely unregulated and as a result does not have standardized requirements of security and privacy practices. In this section, we contribute a list of security and privacy essentials that we believe are basic guarantees. These are not meant to be a comprehensive checklist of features but rather a list of measurable, fundamental aspects that VPN providers should be able to fulfill.

Port Scan: Open ports on a VPN server can be used by malicious actors to identify running services that can be exploited. This measurement scans and discovers the different services running on the VPN server from our custom backend. The application first contacts the backend and establishes a token based on the `UUID`, and the obtained token is then sent to the backend server with a request to scan. Our backend upon receiving the token begins an `Nmap` scan towards the source IP address. In order to restrict the scanning time and target ports of interest, we trigger an `Nmap` scan of a total of 64 (43 TCP, 21 UDP) ports curated using knowledge from the `routersecurity.org` blog. In the *VPN case*, the result of this measurement detects the ports open/filtered and the services running on the VPN server.

Router Scan: This measurement is designed to ascertain if the user's home (ISP) router's management interface is reachable while connected to the VPN. Recent reports have found that multiple models of routers are vulnerable to Remote Code Execution attacks through the router web management interface [11], [12] both pre- and post-authentication. To prevent such web exploits against the user, the router interface should be blocked by the VPN provider.

To detect this, in the bootstrapping stage, we first retrieve the router addresses from the local routing table with platform specific code. In the *VPN case* we run a TCP banner grab, check for HTTP and HTTPS servers on the router's ports 80 and 443 and log the TLS certificate (if available), and send DNS queries to port 53.

DNS Discovery: This measurement is designed to discover all possible DNS resolvers available to the user, separately, in the *VPN case* and *ISP case*. This enables us to check whether the VPN uses either public DNS resolvers or user's ISP resolvers that exposes the users to privacy risks.

In each *VPN case* and *ISP case*, to discover all resolvers, we begin by obtaining the DNS resolvers list from the runtime `Node.js` DNS configuration. We then sequentially query public DNS `whoami` helpers operated by Cloudflare, Akamai, and Google using *each* of the available resolvers. These helpers are designed to respond to 'A' or 'TXT' record queries with the unicast IP address of the recursive resolver that queried the authoritative name server.

Using the discovered resolvers, we do basic liveness checks to ensure they are responsive. These liveness checks are done by sequentially requesting the A, AAAA, SOA, NS records of root name servers. If a resolver is able to obtain a `NOERROR` response for *at least one* of the requests, then it is marked as responsive. This gives us separate lists for *VPN case* and *ISP case* of the responsive DNS resolvers from the user's machine, which is subsequently used for all following DNS-related tests.

Presence of DNS Proxy: This measurement is designed to identify if the VPN has a DNS proxy that targets all the DNS resolvers in the *VPN case*. A DNS proxy is typically used to mitigate DNS leaks and aims to prevent third-parties from potentially monitoring the user's DNS queries. To measure this, we compile a list of public DNS resolvers belonging to Cloudflare, Google, Neustar, OpenDNS, Quad9, VeriSign, and Yandex. Using two randomly selected resolvers from this list, we query the public DNS `whoami` helpers operated by Cloudflare, Akamai, and Google. If the user's VPN connection supports both IPv4 and IPv6, then both endpoints of the `whoami` helpers are queried. This yields the external IP addresses of the DNS resolver(s) making this query.

If the VPN implements a DNS proxy, the ASNs corresponding to the two external IPs should overlap, and the *overlapping ASN should match the ASN of discovered DNS resolvers in VPN case*. Note that we use public DNS resolvers with the assumption that their AS information do not overlap, but even if they do, we do not consider it a DNS proxy unless it also matches the ASN of the resolvers of the *VPN case*. A recent example of public DNS servers' AS overlapping is Neustar's acquisition of Verisign's public DNS service in November of 2020 [62].

Even in the presence of a DNS proxy, the *DNS discovery measurement* can still find DNS leaks if the discovered resolvers belong to an AS other than the VPN's AS. This may happen due to a number of reasons, for instance, in the presence of an IPv6 leak or a malfunctioning DNS proxy.

Support for DNSSEC: This measurement is designed to test if an available resolver validates DNSSEC signatures, a DNS extension to ensure data integrity [35]. For this measurement, we set up a custom authoritative nameserver for a domain under our control. We create a well-configured subdomain with a valid DNSSEC signature, and a subdomain with an invalid DNSSEC signature.

First, the parent zone, say `testdomain.com` is signed, followed by signing of a well-configured subdomain (`good.testdomain.com`) and adding a valid Delegation Signer (DS) record to the parent zone. The DS record is typically generated using the key signing key (KSK) that is created for each signed zone. For `bad.testdomain.com`, we sign the zone correctly but add a malformed DS record to the parent zone. By doing so, the DNSSEC chain of trust is broken between the parent zone and the child zone, and therefore resolvers that validate DNSSEC signatures would throw a SERVFAIL error. Contrarily, a resolver that does not validate DNSSEC signatures would return records successfully for both queries.

Use of Query Name Minimization: This measurement is designed to test if an available resolver implements query name minimization (qmin). Regular DNS queries reveal more information than necessary, and qmin was introduced [37] to limit the query to only include relevant information to a DNS name server. We follow the method introduced by de Vries et al. [13] and set up our own custom test domains. The detection relies on the fact that non-qmin resolvers miss any delegation that happens before the terminal label of a query.

We set up a test domain like `test-qm.testdomain.com` similar to de Vries et al. and set up an authoritative nameserver for that domain. This nameserver *NS* will return a TXT record for the full query `x.y.test-qm.testdomain.com` containing the text "NO, QNAME minimization is not enabled!". But, this nameserver also delegates queries to the second-to-last label, i.e. queries to `y.test-qm.testdomain.com`, to a separate nameserver say *NS**, which when queried for the full domain will return a TXT record containing the text "YES, QNAME minimization is enabled!". Qmin-enabled resolvers will find the record on *NS** whereas non-qmin resolvers that only look for terminal label will find the record on *NS*.

To ensure robustness, we query both our custom domain and the public qmin test service `qnamemintest.internet.nl` and report results based on both queries.

Lack of support for DoH and Presence of DNS64 Resolver: This measurement is designed to test if an available resolver intentionally signals that the network is unsuitable for DNS-Over-HTTPS (DoH). DoH is designed to encrypt the DNS queries to increase user privacy and prevent eavesdropping. Due to the implications on user privacy and discussions surrounding DoH, we are interested in detecting if the VPN and ISP networks disable DoH.

Firefox has now enabled DoH in their browsers by default for US-based users. They use a canary domain [58], namely `use-application-dns.net`, to mitigate compatibility problems for those users who get the DoH feature enabled by default. We include a measurement to query this canary domain and verify the response. A non-`NOERROR` response or the lack of 'A' or 'AAAA' records signals a lack of support for DoH.

A valid exception is when a DNS64 resolver signals lack of support for DoH. NAT64 and DNS64 technologies are generally used by networks to provide IPv4 connectivity for IPv6-only nodes. DNS64 uses "IPv6 address synthesis" to create local IPv6 addresses for IPv4-only services, thereby allowing communication between DNS-using IPv6-only nodes and IPv4-only services. Importantly, DoH standards have declared DNS64 out of their scope. In IPv6-only networks using DNS64/NAT64, third-party DoH is expected to be incompatible and hence would be a viable reason for signalling unsuitability for DoH.

To that end and since such a setup in VPNs is interesting to measure, we test for DNS64 resolvers taking inspiration from RFC 7050 [36], we use the well-known special use IPv4-only domain name `ipv4only.arpa`. All DNS resolvers are supposed to respond with a positive response for a DNS `A` resource record query for the domain and resolve it according to the specification in RFC 7050. In contrast, while requesting a DNS `AAAA` resource record for the same domain, only DNS64 resolvers reply with one or more `AAAA` resource records indicating that they utilize IPv6 address synthesis. Our measurement reports if the available resolvers are DNS64 resolvers and thus signal lack of support for DoH.

TLS Interception: This measurement is designed to identify the presence of TLS interception. VPNs are usually in a privileged position to perform TLS interception as shown by previous work [38]. These interceptions are made possible by the VPN application asking users to install their own, often self-signed, root CA certificates and later generating custom certificates for each TLS connection on the fly. Intuitively, we can measure this by comparing the certificates we fetch through the *VPN case* and the *ISP case*. However, a mismatch does not necessarily mean that the TLS connection has been intercepted because sites like Google or Facebook employ load-balancing with multiple servers containing different certificates.

Taking this factor into account, our *TLS interception test* is designed as follows. We compile a list of three domains—GitHub, Google, and Facebook—that were previously targeted by interception events [41], [78]. These domains were chosen in order to keep the measurement short but at the same time maximize the likelihood of seeing any interception occurrence. We set up a backend at our university and configure it to fetch *all* non-expired certificates for these domains on a daily basis from Certificate Transparency Logs. These certificates are trimmed down to contain only SHA256 signatures and their issuers' CNs, and are updated on our configuration repository daily. During the measurement, these "pre-loaded" certificates are fetched by the application and compared with the certificates we receive from visiting these domains directly. In addition to the three domains mentioned above, we also make two requests to a domain under our control with the SNI header modified to `google.com` or `facebook.com`. This is done based on knowledge from previous work, which showed that adversaries may intercept a connection based on a trigger keyword in the SNI [78]. This measurement identifies TLS Interception if at least one certificate's SHA256 signature and the issuer's CN does not match its counterpart in the pre-loaded set.

TLS Fingerprinting: This measurement is designed to detect the presence of a TLS-terminating endpoint. Previous studies have shown that the wide range of features supported in TLS makes it possible to fingerprint each TLS implementation and to use mismatching fingerprints to detect TLS Man-in-the-middle attacks [21], [50].

To conduct this measurement, we set up a custom fingerprinting backend hosted at our university which listens on TCP port 443. The measurement triggers a TLS handshake with the backend server and sends its experiment `UUID`. Upon receiving a new handshake request, the server will extract the ClientHello from the handshake and hash it to produce a SHA256 fingerprint. We use six fields and extensions of the TLS ClientHello to produce the fingerprint, including server name, supported curves, supported points, signature schemes, supported application protocols, and supported versions. The backend returns this fingerprint back to the application. After querying through both ISP and VPN links, the application will compare the two fingerprints and report any mismatch. A mismatch suggests the presence of a TLS-terminating endpoint in the network path.

VPN Type	Name of VPN Provider
Free Providers (18)	1.1.1.1 + Warp Cloudflare, Best VPN, Free VPN by Free VPN.org, K2VPN, Psiphon, Riseup, Star VPN: Unlimited WiFi Proxy, Touch VPN, Urban VPN desktop, VeePN, VPN Hotspot - Unlimited Proxy, VPN Owl, VPN Plus, VPN Pro, VPN Proxy Master, VPN Super: Best VPN Proxy, VPNBook, VPNLite
Self-hosted (4)	Algo*, OpenVPN Access Server*, Outline*, Streisand*
Paid Providers (58)	AirVPN, Anonine, Astrill VPN*, Atlas VPN *, Avast Secureline, Avira Phantom, AzireVPN, Betternet, BolehVPN, Bullguard*, Cactus VPN, Cryptostorm, CyberGhost *, Encrypt.me *, ExpressVPN *, F-Secure Freedome *, FastestVPN*, Goose VPN*, Hide My Ass! *, Hide.me*, HideIPVPN, Hotspot Shield*, IP Vanish*, IVPN *, Ivacy VPN*, Kaspersky*, KeepSolid/VPN Unlimited *, LeVPN, Mozilla VPN*, Mullvad VPN*, Namecheap*, NordVPN*, Norton Secure VPN*, OVPN.com, PandaVPN, Perfect Privacy *, Private Internet Access*, Private Tunnel, PrivateVPN*, ProtonVPN*, PureVPN*, Speedify *, Steganos, Strong VPN*, SurfEasy, SurfShark *, TorGuard, Trust.Zone*, TunnelBear*, Turbo VPN, University VPN*, Unspyable, VPN.ac, VPNUK, Vypr*, Windscribe*, ZenMate, ZoogVPN

Table II: **VPN list**—Names and types of all 80 tested VPN providers. The 39 VPN providers included in our custom "secure" configurations case study are marked with an asterisk.◇

V. Data Collection

Using *VPNalyzer* for our *data collection*, we aim to investigate VPN providers as well as highlight the value of our application. To demonstrate that *VPNalyzer* works under a variety of conditions, we collect experiments from different types of VPNs (free, paid, and self-hosted) selected for their popularity and market share. These VPNs use different protocols including OpenVPN, Wireguard, IPSec, and IKEv2.

Overall, we have 80 providers: 58 are paid, premium services (includes our University VPN), 18 are free services, and the remaining four are leading self-setup VPN solutions—Outline [40], OpenVPN Access Server [68], Algo [79], and Streisand [77], as illustrated in Table II. The free services are selected from the top free VPNs from the MacOS App Store. For the paid services, we chose the highest tier of service (sometimes called "pro" or "advanced") offered by the provider where applicable.

All 80 providers are tested in their default, "out of the box" security configurations. Since self-hosted VPNs are dependent on client-side software, we tested each of them with their recommended software. We tested the OpenVPN Access Server with the official OpenVPN Connect Client [67], Streisand's OpenVPN offering with Tunnelblick on MacOS and OpenVPN Connect Client on Windows, and Algo and Streisand's Wireguard offering with the official Wireguard application. Outline offers its own client software.

Furthermore, we collect experiments from a subset of the VPN providers by configuring them in a "custom secure mode". VPN providers often offer extra privacy and security functionalities such as blocking third-party DNS, and kill switch. But these features are not *enabled by default* to favor performance or usability. We conduct a case study by zooming

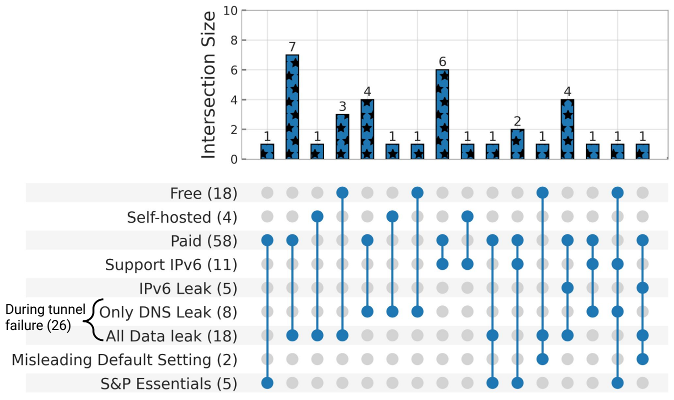


Figure 4: **Summary of Results**—The number of providers with intersecting findings (as indicated by the blue circles) are illustrated. S&P Essentials includes a positive result for DNSSEC, Qmin, RPKI validation, and supports DoH.⋄

All Traffic Leak	Name of VPN Provider
Free Providers (4)	Free VPN by Free VPN.org, Psiphon, Urban VPN desktop, VPN Proxy Master
Self-hosted (1)	OpenVPN Access Server
Paid Providers (8)	Encrypt.me, Hide My Ass!*, IPVanish*, Ivacy VPN, Pure VPN, Speedify, Trust.Zone, Strong VPN*
Paid & Leaks IPv6 (5)	Astrill VPN*, Norton Secure VPN, SurfEasy, Turbo VPN, University VPN
Only leaks DNS traffic during tunnel failure (8)	1.1.1.1+Warp, Avira Phantom VPN, Betternet, Hotspot Shield*, Private Internet Access*, Streisand (on OpenVPN Connect v3), TunnelBear, VPN Owl

Table III: **Providers with traffic leakages**—26 providers leak traffic during tunnel failure. * indicates those with traffic leaks even when their "kill switch" feature is enabled, see §VI-H.⋄

in on the 34 most popular VPN providers (all paid services) selected by combining the top VPN recommendation sites based on search engine results, listed in Table V. We augmented this list with the four self-hosted VPN solutions and our institutional university VPN. For this case study, we use custom secure configurations where we enabled all relevant security and privacy settings available on the VPN client software. We repeat our testing on both Windows and MacOS (not all VPN providers had client software for Linux). We report our findings from this case study in §VI-H.

In total, including our case study, we analyzed 230 experiments from 80 VPN providers, all run before July 24, 2021. The names of all the 80 providers is presented in Table II.

VI. Findings

VPNalyzer uncovers several previously unreported issues from the 80 tested VPN providers. We note that this is only a snapshot of the tested VPN providers. There have already been changes in ownership, and operation of some providers between the time of testing and publication, and the fact that the VPN ecosystem is constantly evolving and changing highlights the need for a tool such as *VPNalyzer* to continue to test and monitor them. We have responsible disclosures in progress regarding our findings.

In this section, we describe our findings in-depth and extract key takeaways. Moreover, considering that the implications of our findings may vary based on the readers' threat model, we provide a condensed visualization in Figure 4 and our raw findings in Table VI in the Appendix. For instance, if a reader wants to identify how many tested paid providers have traffic leaks during tunnel failure and do not satisfy *some* security and privacy essentials, the second intersection in Figure 4 shows the answer is seven paid VPN providers.

A. Support for IPv6

Majority of VPN providers and servers do not support IPv6. Alarmingly, five VPN providers do not block IPv6 traffic thereby leaking user data to their ISP IPv6 link

Overall, we are the first to show that a majority of VPN providers do not offer support for IPv6—only 11 providers out of 80 tested had IPv6 connectivity. Our data collection consists of multiple experiments for a majority of the VPN providers, and we note that no provider had explicit labelling of servers having IPv6 capability. The state of IPv6 adoption around the world has been increasing steadily, with ≈30% of Alexa Top 1,000 sites being reachable over IPv6 [84] and over 36% of Google users accessing their services over IPv6 as of July 2021 [25]. However, this trend is not reflected in the VPN providers tested.

Alarmingly, our *AS Mismatch* measurement discovered that five VPN providers leak IPv6 traffic. These VPN providers—Astrill VPN, Norton Secure VPN, Turbo VPN, SurfEasy VPN, and our university VPN—do not block the user's IPv6 connectivity and thus *leak IPv6 data to the ISP*. We filed responsible disclosures for this issue to the providers, and our university VPN has already fixed this IPv6 leak by blocking IPv6 connectivity.

B. Traffic Leakages

Twenty-six VPN providers leak user data to the ISP during tunnel failure; eight of which leak DNS traffic alone, and the remaining 18 in addition leak other types of traffic

Our *VPN kill switch test* discovers 18 VPN providers leak all user traffic during tunnel failure, suggesting a missing kill switch feature or a faulty implementation. Out of these 18, four are free VPN providers, 13 are paid providers, and one is self-hosted, as reported in Table III. The 13 paid providers also include five that leak IPv6 traffic, described previously in §VI-A. The kill switch feature is commonly disabled by default in the VPN application often citing that it interferes with the user's browsing experience and lowers usability. In case of tunnel failure, a missing or misconfigured kill switch can lead to privacy and security issues such as leaking sensitive user data. Unfortunately, these tunnel failures are easy to induce, e.g. by simply dropping packets to VPN server, especially by active adversaries such as governments and ISPs.

Our *DNS leak during tunnel failure* measurement finds 26 providers leaking DNS traffic, which includes the above 18, and eight other providers including top ones such as TunnelBear and Private Internet Access. As a result of extensively studying different kill switch implementations, we are the first to measure and discover DNS leaks during tunnel failure which occurs due to VPN providers allowing DNS queries to bypass the tunnel, possibly in an effort to facilitate reconnection. This implementation decision opens up a serious vulnerability which can be avoided in a plethora of ways such as diagnosing issues with and reordering the machine's firewall, and caching VPN-related DNS names to attempt reconnection.

Of the eight providers that leak only DNS traffic, one is a self-hosted VPN–Streisand's OpenVPN configuration on Windows, two are free providers, and five are paid providers. Note that our measurement only reports "definitive" leaks, and is a conservative lower bound. Nevertheless, our findings show that a significant number of popular VPN providers are affected by leakages during tunnel failure, potentially exposing user data to adversaries.

C. Security and Privacy Essentials

Adoption of practices we consider security and privacy essentials is not uniform across VPN providers

Findings from our *Support for DNSSEC* and *Use of Query Name Minimization* measurements show that many VPN providers do not implement DNS security and privacy measures. We find that only 54 VPN providers use resolvers that have DNSSEC validation enabled, which is important to ensure DNS data integrity. We find that an even fewer number of VPN providers, only 26, use resolvers that support query name minimization(qmin), which was specifically introduced in RFC 7816 to improve DNS privacy.

From our *RPKI validation* measurement, we find that 35 VPN providers have servers in networks where the RPKI validation is enabled. Note that since VPN servers are usually distributed widely in different networks, our result only means that at least one of the tested servers of the provider validates RPKI. Although the proportion of providers performing RPKI validation is higher than Cloudflare's global estimates (20% of Internet) in late 2020 [8], VPN providers should be ahead of the curve especially to protect against issues like BGP hijacking.

We also discover that 14 VPN providers have configured their network to disable DNS-over-HTTPS for Firefox users via their canary domain [58] without the presence of a DNS64 resolver. The Mozilla canary domain is a deliberate signal to convey that the network is unsuitable for DoH, and will result in DoH being disabled for those users who had it enabled by default on Firefox. We believe that doing so without reason, or without informing users is poor practice [57], [59]. To learn if they have any operational reasons, we contacted these 14 VPN providers as part of our responsible disclosure.

On a positive note, from our *Router Scan* measurement, we find that five VPN providers—Cactus VPN, Encrypt.me, IVPN, Mullvad VPN, and Windscribe—block access to the local ISP router interface while connected to the VPN server. For instance, Mullvad VPN always blocks traffic going to and from the ISP router (and local network) while connected. We recommend that all VPN providers block access to the router interface to protect their users from potentially being vulnerable to a class of deanonymization attacks through the router management web interface and/or the stub DNS resolver.

From our *Port Scan* measurement, we see that VPN servers have port 443 open in 46 VPN providers, port 80 open in 23 VPN providers, ports 53 and 8443 open in 11 providers each, and port 8080 open in 10 providers. While these are not necessarily security risks, they can be used by malicious actors to identify and conduct active probing against VPN servers and can be exploited in a number of ways [32].

In all of the above findings, our results for a particular provider over different experiments and servers in Windows and MacOS are highly concordant.

IP Block	VPN Providers
AS 9009:	
37.120.128.0/17	IPVanish, Touch VPN, Nord VPN, Norton Secure VPN
217.138.192.0/18	Boleh VPN, Ivacy VPN, Hide.me, Mozilla VPN, Goose VPN, Windscribe
5.181.234.0/24	Pure VPN, OVPN
95.174.64.0/22	Fastest VPN, Atlas VPN
AS 60068:	
84.17.48.0/21	CyberGhost VPN, NordVPN

Table IV: **VPN Providers with shared infrastructure**—IP blocks shared by different providers in AS 9009 and 60068.◇

D. Sharing of VPN Infrastructure

Many VPN providers use the same underlying infrastructure

Using our *AS Mismatch* measurement, we find that the servers of 27 VPN providers belong to a single AS (AS 9009-M247 Ltd). While previous work found one shared IP block in this AS [43], *VPNalyzer* finds that 14 VPN providers share four IP blocks listed in Table IV. The other 13 providers' servers were distributed across the IP space belonging to AS 9009. Additionally, we find an IP block shared by two providers in AS 60068 (Datacamp Limited). Such shared IP blocks are easier for censors or other adversaries to identify and block.

Colocation of VPN servers could be the result of bilateral partnerships [43], such as Mozilla VPN using servers "powered by Mullvad" [60]. From our measurement, we find that IP blocks in AS 16509 (Amazon) infrastructure are shared across Norton Secure VPN and SurfEasy VPN, which are both different brands under the NortonLifeLock Product family [64]. Our results are only a lower bound, as we did not connect to every single server available from each provider, since studying this was not our main goal. Furthermore, sharing of infrastructure could also be due to small VPN providers outsourcing or renting resources from a large cloud/hosting provider, or a single parent company owning multiple VPN brands that share infrastructure [80].

E. Deceptive and/or Malicious Behavior

Malicious and deceptive behaviors such as traffic interception are not widespread but are not non-existent

We find that malicious behavior such as TLS interception is less widespread, to an even lesser extent than previously reported [38], [43]. This could be due to the fact that we test more popular, premium VPN providers that are used by a large population of users. Even so, we do still find evidence of manipulation, previously unreported, in at least two providers—Betternet (on both MacOS and Windows) and Turbo VPN (Windows)—as well as deceptive geolocation claims by four free providers.

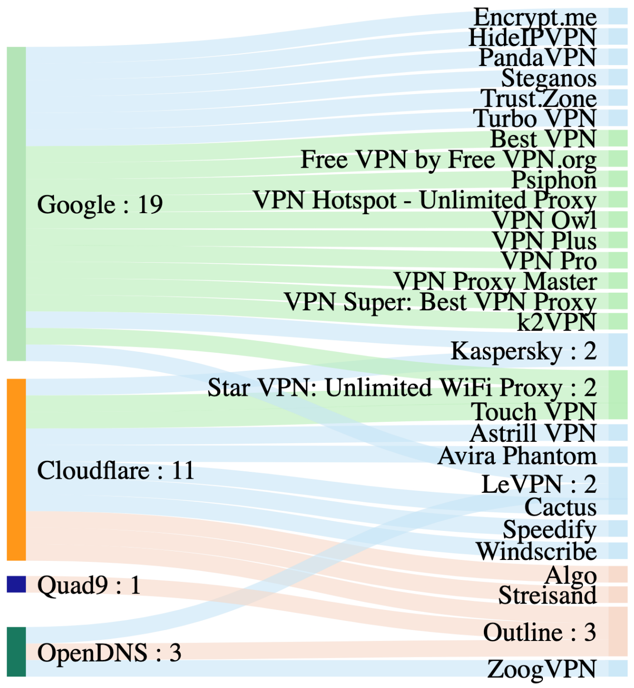


Figure 5: **Use of Public DNS Services**—List of 29 VPN providers that use popular public DNS services. Free VPNs are colored in green, paid in blue, and self-hosted in orange.◇

We find two instances of abnormal TLS responses and one provider serving non-genuine response for DNS queries. Betternet returns an RFC 6598 Carrier-grade NAT address for all DNS queries. Although this could be a design choice for optimization, we still label it unexpected behavior. In our *TLS interception* measurement, when we request our custom domain with the SNI header modified to `google.com`, we see that both Betternet and Turbo VPN return the certificate belonging to Google whereas we expect to see our custom domain's certificate. This could be due to an internal policy of handling requests to entities such as Google [19] but we still report it as abnormal behavior. We have reached out to the providers as part of our responsible disclosure.

From our *Geolocation test*, we find instances of "deceptive" geolocation in four out of the 18 free VPNs—namely Free VPN by Free VPN.org, VPN Owl, VPN Hotspot - Unlimited Proxy, and VPN Super. In these providers, the VPN-advertised server location and the geolocation determined by our Cloudflare endpoint do not match. They range from nearby countries (United Kingdom and Germany), to different hemispheres (Japan and Australia). This corroborates findings from Weinberg et al. in [82] and has potential for further future work using the active geolocation measurement data collected using *VPNalyzer*.

F. Use of Public DNS Services

Twenty-nine VPN providers (including paid, premium ones) configure clients to use public DNS servers

From our *DNS Discovery* measurement, we observe that 29 VPN providers have configured their client applications to use public DNS resolution services, such as Google Public DNS, Cloudflare DNS, OpenDNS, and Quad9, as illustrated in Figure 5. Users must be informed that their DNS queries are being routed to third-party organizations such as Google in addition to their VPN provider, especially when users pay for the VPN service [73], [75]. In general, our goal is to report to the user all the entities that handle their DNS queries. Depending on the user's threat model and VPN use case, this information may be critical.

Of these 29 providers, we find two interesting behaviors. We observe that Windscribe uses its own DNS servers, but from our *DNS discovery* we see that when the recursion is done over IPv6, it uses Cloudflare DNS. On the other hand, Trust.zone always uses Google Public DNS on Windows but allows a fallback to using user's default DNS through the VPN on MacOS due to a configuration failure on Tunnelblick.

Among the free VPN providers, 12 out of 18 configure their users to use public DNS servers with Google Public DNS being the most popular choice, followed by Cloudflare DNS. Additionally, we find that three of the self-hosting VPN solutions, Algo, Streisand, and Outline configure the client to use public DNS services. In the case of the self-hosted solutions, having no option to easily override the use of the public DNS services could prove problematic in countries where access to these public DNS services is blocked.

Some VPN providers host their "own" DNS service on hosting providers such as Amazon, Linode, and Digital Ocean, which is different from where their corresponding VPN servers are hosted. While this is different from simply outsourcing user queries to public DNS services, we still note that users' DNS queries are being served through different hosting providers.

G. Use of DNS proxies and Mitigating DNS Leaks

Twelve VPN providers have implemented DNS proxies which help mitigate DNS leaks

From our *DNS Proxy* measurement, we find 12 VPN providers employing custom DNS proxies in their network in order to mitigate DNS leaks. On a positive note, of these 12, we also find some VPN providers such as Mullvad VPN, explicitly blocking queries to public DNS resolvers and only allowing queries sent to their own DNS servers. We believe that this is a good security practice, as it is done in an attempt to ensure that the user's DNS queries are not leaked to other third-party DNS resolvers.

H. Case Study: Testing Custom Secure Configurations

Ten VPN providers out of 39 tested leak traffic even in a more secure setting, six even had a "kill switch" enabled

VPN providers often optimize for performance and usability over security because features like blocking third-party DNS and kill switch can interfere with user experience. To measure if VPN providers offer meaningful, configurable security features in their application, we conduct this case study of testing the 39 most popular VPN providers and find that a multitude of issues persist even in the secure mode.

From our measurements, we find traffic leaks in 10 providers tested in their secure mode, six of which *even had a kill switch setting enabled*, as detailed in Table III. Furthermore, we also observe two providers—Astrill VPN, Norton Secure VPN—which were reported in §VI-A to leak IPv6 traffic, also do so in their most secure configuration. This case study highlights egregious implementation failures as these 10 providers suffer from information leakage during tunnel failure even in their most secure configuration, shown in Table VI. While all 10 providers in their secure configuration, including top providers like TunnelBear, leak DNS traffic during tunnel failure, six of them *also leak all traffic*.

We find that IPv6 support is an "advanced" feature that needs to be turned on by the user in at least two providers—IVPN and Mullvad VPN. Furthermore, we noticed that VPN providers have different names for their mechanism to protect users' data from leaking during tunnel failure (e.g. VigilantBear, Kill switch, Disable network access, Firewall always on *etc*), making it difficult for users to identify and enable this feature.

We find that four VPN providers—Bullguard VPN, F-secure Freedome, KeepSolid, and ProtonVPN—block access to ISP routers *only in the "secure" mode*. For example, ProtonVPN allows access to the ISP router interface under the default configuration but blocks the access when "Netshield" is enabled.

Alarmingly, we find that two of the 80 VPN providers have misleading default settings. The Astrill VPN and Psiphon applications are configured to tunnel only browser traffic *by default* and hence, for all our findings in the paper, Astrill VPN was run using both its OpenVPN and Wireguard protocol (also part of this case study in its secure modes), and Psiphon was run using its L2TP/IPsec protocol. We note that their default configuration is unsafe, as it poses potential security and privacy risks for users who may transmit sensitive information using non-browser apps, under the assumption that the VPN application encrypts all their traffic. Typically, VPN products that tunnel only browser traffic are downloaded and configured as browser extensions. Given that users have to download an app to install and run Astrill VPN and Psiphon, it may mislead them to think that all traffic is tunneled by default.

VII. Discussion

Our evaluation of 80 popular VPN providers with *VPNalyzer* uncovers several important issues with VPN providers and shows the usefulness of *VPNalyzer* as a tool to investigate the VPN ecosystem at scale. While one experiment on *VPNalyzer* currently takes ≈20 minutes to run and conducts 15 measurements, the modular design supports the efficient addition of new tests and upgrades to existing measurements. *VPNalyzer* can be updated continuously to address the evolving nature of threats in the VPN ecosystem. Although we do not focus on testing all available servers for each VPN provider, our findings are highly consistent across the multiple servers tested for a provider. We highlight the results of most measurements in §VI, but we intentionally choose not to report the bandwidth measurements collected by *VPNalyzer*. While the individual bandwidth results are interesting to users and are displayed on the app, a large-scale comparison of bandwidth between VPN providers is subject to many compounding factors, such as time of measurement and traffic capacity of ISPs.

Our findings shed light on the lack of standardization and regulation by highlighting the varying levels of security and privacy we see offered by the VPN providers. We discover that the mechanism to protect user's traffic during tunnel failure (i.e. kill switch, firewalls, shields), IPv6 connectivity, blocking of ads and tracking, and smart/secure DNS are all features often disabled in the "default" mode of VPN applications. While we recognize that this is a conscious decision from the VPN provider, these settings should be made more accessible and user-friendly. There is no standard jargon for these features, and the names and capabilities of simple settings are often exaggerated by VPN providers. For instance, terms like "military grade encryption", and "smart connect" are often proffered with little explanation accompanying them.

There have been attempts by certain VPN providers to form coalitions like the VPN Trust Initiative (VTI) [34] that take steps towards regulation and setting industry best practices. However due to vested interests of a handful of VPN providers, these efforts are typically not adopted by other popular providers in the VPN ecosystem. Moreover, anecdotal evidence suggests that VPNs in these coalitions do not follow all its basic principles. For instance, Ivacy VPN which is part of VTI, advertises "anonymity" prominently on its website [39] whereas "Never claim VPNs guarantee anonymity" is one of the basic principles of the coalition [34].

There is a need for independent, unbiased parties to put forward standards based on systematic, data-driven studies such as the one provided in this paper. To that end, we partnered with *Consumer Reports* and served as panelists on a workshop about VPNs organized by them which was attended by over 1,500 users. Thereafter, *Consumer Reports* used our *VPNalyzer* tool as the first in a line of systematic investigation to help evaluate a set of popular providers for a recommendation article on their website. The simplicity and usability of our *VPNalyzer* tool helped *Consumer Reports* to also test providers to evaluate and recommend.

Responsible disclosure of issues found by *VPNalyzer* are already helping VPN providers improve and fix issues with their service. For instance, our university VPN has already fixed their IPv6 leak, and made their kill switch implementation more secure. For our next steps, we plan to release the *VPNalyzer* tool for a wider audience in the coming months and we hope that *VPNalyzer* benefits users and helps the general public choose better VPN providers.

VIII. Conclusion

To our knowledge, we are the first research study to build a system, *VPNalyzer*, and develop a tool with automated tests and functionality to empower researchers and users to asses the service provided to them by VPN providers. We demonstrate that the *VPNalyzer* application is able to find important issues by analyzing 230 experiments from 80 popular and diverse desktop VPN providers. We find several notable issues and vulnerabilities, including IPv6 leaks, kill switch leaks, DNS leaks during tunnel failure, and the lack of adoption of certain security and privacy essentials. We plan to conduct a public release of the *VPNalyzer* tool in the coming months, which will only further increase our measurement scope and findings. We hope that *VPNalyzer* benefits researchers and users alike, helps the general public make more-informed decisions about which providers to use for their particular needs, and ultimately fosters stronger security and privacy practices in the VPN ecosystem.

Acknowledgment

The authors are grateful to Ben Stock for shepherding the paper, and to Adam Lynn, David Fifield, Ben Moskowitz, Leah Fischman, Yael Grauer, and the reviewers for their feedback. This work was made possible by the Open Technology Fund, and the Consumer Reports Digital Lab Fellowship.

References

[1] D. Allan, “CyberGhost owner buys PIA for $95.5m to create VPN giant,” TechRadar, 2019, https://www.techradar.com/news/cyberghost-owner-buys-pia-for-dollar955m-to-create-vpn-giant.

[2] ARIN, “Resource Certification (RPKI),” 2021, https://www.arin.net/resources/manage/rpki/.

[3] ARIN, “Route Origin Authorizations (ROAs),” 2021, https://www.arin.net/resources/manage/rpki/roa_request.

[4] S. Banjo, “VPN Downloads Surge in Response to Hong Kong Security Law,” Bloomberg, 2020, https://www.bloomberg.com/news/articles/2020-05-22/vpn-downloads-surge-in-response-to-hong-kong-security-law.

[5] T. Bui, S. Rao, M. Antikainen, and T. Aura, “Client-Side Vulnerabilities in Commercial VPNs,” in *Nordic Conference on Secure IT Systems*. Springer, 2019.

[6] C. Chen, “America has killed Net Neutrality; Why you need a VPN service,” Private Internet Access, 2018, https://www.privateinternetaccess.com/blog/america-has-killed-net-neutrality-why-you-need-a-vpn-service/.

[7] C. Cimpanu, “Data breach at Russian ISP impacts 8.7 million customers,” ZDNet, 2019, https://www.zdnet.com/article/data-breach-at-russian-isp-impacts-8-7-million-customers/.

[8] Cloudflare, “The Internet is Getting Safer: Fall 2020 RPKI Update,” 2020, https://blog.cloudflare.com/rpki-2020-fall-update.

[9] Cloudflare, “Configuring Cloudflare IP Geolocation,” 2021, https://web.archive.org/web/20210816060903/https://support.cloudflare.com/hc/en-us/articles/200168236-Configuring-Cloudflare-IP-Geolocation.

[10] Cloudflare, “How VPNs affect Internet speed,” 2021, https://www.cloudflare.com/learning/access-management/vpn-speed.

[11] “CVE-2019-12103,” https://nvd.nist.gov/vuln/detail/CVE-2019-12103.

[12] “CVE-2019-19356,” https://nvd.nist.gov/vuln/detail/CVE-2019-19356.

[13] W. B. de Vries, Q. Scheitle, M. Müller, W. Toorop, R. Dolmans, and R. van Rijswijk-Deij, “A First Look at QNAME Minimization in the Domain Name System,” in *International Conference on Passive and Active Network Measurement*. Springer, 2019.

[14] L. DiCioccio, R. Teixeira, and C. Rosenberg, “Measuring home networks with homenet profiler,” in *International Conference on Passive and Active Network Measurement*. Springer, 2013.

[15] D. Dittrich and E. Kenneally, “The Menlo Report: Ethical principles guiding information and communication technology research,” U.S. Department of Homeland Security, Tech. Rep., 2012.

[16] Electron, “Build cross-platform desktop apps with JavaScript, HTML, and CSS,” 2021, https://www.electronjs.org/.

[17] S. Englehardt and A. Narayanan, “Online tracking: A 1-million-site measurement and analysis,” in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016.

[18] L. Fazal, S. Ganu, M. Kappes, A. S. Krishnakumar, and P. Krishnan, “Tackling security vulnerabilities in VPN-based wireless deployments,” in *Proceedings of IEEE International Conference on Communications, ICC 2004, Paris, France, 20-24 June 2004*, 2004.

[19] D. Fifield, J. Jian, and P. Pearce, “SNI proxies,” Bamsoftware, 2016, https://www.bamsoftware.com/computers/sniproxy/index.html.

[20] A. Filasto and J. Appelbaum, “OONI: Open Observatory of Network Interference,” in *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2012.

[21] S. Frolov and E. Wustrow, “The use of TLS in censorship circumvention.” in *Network and Distributed Systems Symposium (NDSS)*, 2019.

[22] D. Gewirtz, “Best VPN services for 2021: Safe and fast don’t come for free,” ZDNet, 2021, https://www.zdnet.com/article/best-vpn-services-for-2021-safe-and-fast-dont-come-for-free.

[23] M. Gharaibeh, A. Shah, B. Huffaker, H. Zhang, R. Ensafi, and C. Papadopoulos, “A look at router geolocation in public and commercial databases,” in *Proceedings of the 2017 Internet Measurement Conference*, 2017.

[24] Google, “Google Cloud: Cloud Computing Services,” 2021, https://cloud.google.com.

[25] Google, “Google IPv6 Statistics,” 2021, https://www.google.com/intl/en/ipv6/statistics.html.

[26] Y. Grauer, “The Best VPN Service,” New York Times Wirecutter, 2020, https://www.nytimes.com/wirecutter/reviews/best-vpn-service/.

[27] Y. Grauer, “Should You Use a VPN?” in *Consumer Reports Digital Lab*, 2021, https://www.consumerreports.org/vpn-services/should-you-use-a-vpn-a5562069524/.

[28] Y. Grauer, “VPN Testing Reveals Poor Privacy and Security Practices, Hyperbolic Claims,” in *Consumer Reports Digital Lab*, 2021, https://www.consumerreports.org/vpn-services/vpn-testing-poor-privacy-security-hyperbolic-claims-a1103787639.

[29] Y. Grauer and S. Blair, “Security and Privacy of VPNs Running on Windows 10,” in *Consumer Reports Digital Lab*, 2021, https://digital-lab-wp.consumerreports.org/wp-content/uploads/2021/12/VPN-White-Paper.pdf.

[30] A. Hochstadt, “The death of net neutrality and the rise of VPNs,” TechRadar, 2018, https://www.techradar.com/news/the-death-of-net-neutrality-and-the-rise-of-vpns.

[31] R. Hodge, “VPN use surges during the coronavirus lockdown, but so do security risks,” CNET, 2020, https://www.cnet.com/news/vpn-use-surges-during-the-coronavirus-lockdown-but-so-do-security-risks/.

[32] M. Horowitz, “RouterSecurity.org - TCP Ports to Test,” https://routersecurity.org/testrouter.php#TCPports.

[33] D. Y. Huang, N. Apthorpe, F. Li, G. Acar, and N. Feamster, “IoT Inspector: Crowdsourcing Labeled Network Traffic from Smart Home Devices at Scale,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2020.

[34] i2Coalition, “The VPN Trust Initiative ("VTI"),” https://vpntrust.net/, September 2020.

[35] IETF, “RFC 4033: DNS Security Introduction and Requirements,” 2005, https://datatracker.ietf.org/doc/html/rfc4033.

[36] IETF, “RFC 7050: Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis,” 2013, https://datatracker.ietf.org/doc/html/rfc7050.

[37] IETF, “DNS Query Name Minimisation to Improve Privacy,” 2016, https://tools.ietf.org/html/rfc7816.

[38] M. Ikram, N. Vallina-Rodriguez, S. Seneviratne, M. A. Kaafar, and V. Paxson, “An Analysis of the Privacy and Security Risks of Android VPN Permission-enabled Apps,” in *Proceedings of the 2016 Internet Measurement Conference*, 2016.

[39] Ivacy VPN, “Ivacy VPN - Stay Anonymous Online with a VPN!” https://web.archive.org/web/20210716224231/https://www.ivacy.com/what-is-vpn/anonymous-vpn/.

[40] Jigsaw, “Outline VPN,” https://getoutline.org.

[41] M. Johnson, “China, GitHub and the man-in-the-middle,” GreatFire.org, January 30, 2013, https://en.greatfire.org/blog/2013/jan/china-github-and-man-middle.

[42] J. Kastrenakes, “NordVPN reveals server breach that could have let attacker monitor traffic,” The Verge, 2019, https://www.theverge.com/2019/10/21/20925065/nordvpn-server-breach-vpn-traffic-exposed-encryption.

[43] M. T. Khan, J. DeBlasio, G. M. Voelker, A. C. Snoeren, C. Kanich, and N. Vallina-Rodriguez, “An Empirical Analysis of the Commercial VPN Ecosystem,” in *Proceedings of the 2018 Internet Measurement Conference*, 2018.

[44] T. Khan, “VPN Tests,” GitHub, 2018, https://github.com/tahakhan5/vpn_tests/blob/master/leakage_tests/tunnel_failure/run_test.py#L60.

[45] C. Kreibich, N. Weaver, G. Maier, B. Nechaev, and V. Paxson, “Experiences from Netalyzr with engaging users in end-system measurement,” in *Proceedings of the first ACM SIGCOMM workshop on Measurements up the stack*, 2011.

[46] C. Kreibich, N. Weaver, B. Nechaev, and V. Paxson, “Netalyzr: Illuminating the edge network,” in *Proceedings of the 2010 ACM SIGCOMM Conference on Internet Measurement*, 2010.

[47] Lai Yi Ohlsen, Matt Mathis, Stephen Soltesz, Simone Basso, “Introducing ndt7,” 2020, https://www.measurementlab.net/blog/ndt7-introduction/.

[48] F. Li, A. A. Niaki, D. Choffnes, P. Gill, and A. Mislove, “A large-scale analysis of deployed traffic differentiation practices,” in *Proceedings of the ACM Special Interest Group on Data Communication*. SIGCOMM, 2019.

[49] “Lumen Privacy Monitor,” https://www.icsi.berkeley.edu/icsi/projects/networking/haystack.

[50] “Measuring Active Listeners, Connection Observers, and Legitimate Monitors, Cloudflare,” https://malcolm.cloudflare.com/.

[51] P. Matic, “VPN: A Decade’s Worth of Growth,” 2020, https://www.pcmatic.com/news/vpn_report/.

[52] E. Max, “The Best VPN Services for 2021,” PCMag, 2021, https://www.pcmag.com/picks/the-best-vpn-services.

[53] N. McCarthy, “VPN Usage Surges During COVID-19 Crisis,” Forbes, 2020, https://www.forbes.com/sites/niallmccarthy/2020/03/17/vpn-usage-surges-during-covid-19-crisis-infographic/?sh=5fe8e5157d79.

[54] A. McDonald, M. Bernhard, L. Valenta, B. VanderSloot, W. Scott, N. Sullivan, J. A. Halderman, and R. Ensafi, “403 Forbidden: A global view of CDN geoblocking,” in *Proceedings of the 2018 Internet Measurement Conference*, 2018.

[55] “Measurement Lab,” https://www.measurementlab.net/tests/ndt/.

[56] A. Molavi Kakhki, A. Razaghpanah, A. Li, H. Koo, R. Golani, D. Choffnes, P. Gill, and A. Mislove, “Identifying traffic differentiation in mobile networks,” in *Proceedings of the 2015 Internet Measurement Conference*, 2015.

[57] Mozilla, “What’s next in making Encrypted DNS-over-HTTPS the Default,” Mozilla Blog, 2019, https://blog.mozilla.org/futurereleases/2019/09/06/whats-next-in-making-dns-over-https-the-default/.

[58] Mozilla, “Canary domain - `use-application-dns.net`,” 2021, https://support.mozilla.org/en-US/kb/canary-domain-use-application-dnsnet.

[59] Mozilla, “Can’t networks just trigger the canary domain check all the time and disable DoH?” Mozilla Support, 2021, https://support.mozilla.org/en-US/kb/dns-over-https-doh-faqs#w_cant-networks-just-trigger-the-canary-domain-check-all-the-time-and-disable-doh.

[60] Mozilla, “Mozilla VPN,” 2021, https://web.archive.org/web/20210531041656/https://www.mozilla.org/en-US/products/vpn/.

[61] Namecheap, “Google Trends Reveals Surge in Demand for VPN,” https://www.namecheap.com/blog/vpn-surge-in-demand/, 2020.

[62] Neustar, “Neustar Announces Acquisition of Verisign’s Public DNS Service,” 2020, https://www.home.neustar/about-us/news-room/press-releases/2020/neustar-announces-acquisition-of-verisigns-public-dns-service.

[63] Nmap Project, “Npcap: Packet capture library for Windows,” https://nmap.org/npcap.

[64] “NortonLifeLock Product and Services,” https://web.archive.org/web/20210719021634/https://www.nortonlifelock.com/us/en/privacy/product-privacy-notices/.

[65] OONI, “OONI Data Policy,” 2020, https://ooni.org/about/data-policy.

[66] OpenVPN, “Reference manual for OpenVPN 2.4,” 2021, https://openvpn.net/community-resources/reference-manual-for-openvpn-2-4/.

[67] OpenVPN Inc., “Official OpenVPN Connect client software,” https://openvpn.net/vpn-client/.

[68] OpenVPN Inc., “OpenVPN Access Server,” https://aws.amazon.com/marketplace/pp/prodview-fiozs66safl5a.

[69] S. Park and K. Albert, “A Researcher’s Guide to Some Legal Risks of Security Research,” 2020, https://clinic.cyber.harvard.edu/files/2020/10/Security_Researchers_Guide-2.pdf.

[70] V. C. Perta, M. V. Barbera, G. Tyson, H. Haddadi, and A. Mei, “A glance through the VPN looking glass: IPv6 leakage and DNS hijacking in commercial VPN clients,” *Proceedings on Privacy Enhancing Technologies*, 2015.

[71] I. Poese, S. Uhlig, M. A. Kaafar, B. Donnet, and B. Gueye, “IP geolocation databases: Unreliable?” *ACM SIGCOMM Computer Communication Review*, 2011.

[72] Prometheus, “Prometheus - Monitoring system & time series database,” 2021, https://prometheus.io/.

[73] A. Randall, E. Liu, G. Akiwate, R. Padmanabhan, G. M. Voelker, S. Savage, and A. Schulman, “Trufflehunter: Cache snooping rare domains at large public dns resolvers,” in *Proceedings of the 2020 ACM Internet Measurement Conference*, 2020.

[74] RIPE, “RIPEstat Data API,” 2021, https://stat.ripe.net/docs/data_api.

[75] K. Schomp, M. Allman, and M. Rabinovich, “DNS resolvers considered harmful,” in *Proceedings of the 13th ACM Workshop on Hot Topics in Networks*, 2014.

[76] M. Snider, “ISPs can now collect and sell your data: What to know about Internet privacy rules,” USA TODAY, 2017, https://www.usatoday.com/story/tech/news/2017/04/04/isps-can-now-collect-and-sell-your-data-what-know-internet-privacy/100015356/.

[77] StreisandEffect, “Streisand,” https://github.com/StreisandEffect/streisand.

[78] R. Sundara Raman, L. Evdokimov, E. Wurstrow, J. A. Halderman, and R. Ensafi, “Investigating Large Scale HTTPS Interception in Kazakhstan,” in *Proceedings of the 2020 ACM Internet Measurement Conference*, 2020.

[79] trailofbits, “Algo VPN,” https://github.com/trailofbits/algo.

[80] VPNMentor, “These 7 Companies Secretly Own Dozens of VPNs,” 2021, https://www.vpnmentor.com/blog/companies-secretly-own-dozens-vpns.

[81] N. Weaver, C. Kreibich, and V. Paxson, “Redirecting DNS for Ads and Profit,” in *USENIX Workshop on Free and Open Communicationson the Internet (FOCI)*, 2011.

[82] Z. Weinberg, S. Cho, N. Christin, V. Sekar, and P. Gill, “How to catch when proxies lie: Verifying the physical locations of network proxies with active geolocation,” in *Proceedings of the 2018 Internet Measurement Conference*, 2018.

[83] Wireguard, “Endpoint with changing IP,” ArchLinux, 2021, https://wiki.archlinux.org/title/WireGuard#Endpoint_with_changing_IP.

[84] “World IPv6 Launch,” 2021, https://www.worldipv6launch.org/measurements/.

Appendix

A. Recommendation Websites

Table V shows the 25 recommendation websites used to create the set of popular VPN providers, which we augment with the four self-hosted VPNs and our institutional VPN.

Recommendation Websites Used
https://www.security.org/vpn/best/
https://www.techradar.com/vpn/best-vpn
https://www.cnet.com/news/best-vpn/
https://www.tomsguide.com/best-picks/best-vpn
https://www.pcmag.com/picks/the-best-vpn-services
https://thebestvpn.com/
https://www.wired.co.uk/article/best-vpn
https://www.zdnet.com/article/best-vpn/
https://www.cloudwards.net/best-vpn/
https://www.internetsecurity.org/compare/usa
https://www.top10vpn.com/best-vpn-for-usa/v/d/?bsid=c33se1kw011
https://bestvaluevpn.com/usd/best-vpn/?utm_campaign=ggls-en-usa-gen
https://www.nytimes.com/wirecutter/reviews/best-vpn-service/
https://cybernews.com/best-vpn/
https://vpnoverview.com/best-vpn/top-5-best-vpn/
https://www.guru99.com/best-vpn-usa.html
https://www.crazyegg.com/blog/best-vpn-services/
https://www.forbes.com/advisor/business/software/best-vpn/
https://blog.flashrouters.com/vpn/
https://vpnpro.com/best-vpn-services/
https://bestvpn.org/best-vpns-for-the-usa/
https://www.safetydetectives.com/best-vpns (formerly thatoneprivacyguy)
https://www.tomsguide.com/best-picks/best-free-vpn
https://www.top50vpn.com/best-vpn
https://www.top10vpn.com/best-vpn/
Total: 25

Table V: **Websites Used**—The top 25 websites used to create a set of 34 of the most popular VPN providers. Buffered VPN was among this list but is currently non-operational.◇

B. Full Results

Table VI shows a complete summary of our findings.

			Leaks during tunnel failure		Security & Privacy Essentials					
#	VPN Provider	IPv6 Support	All Data Leak	DNS Leak	DNSSEC Validation	Qmin Support	RPKI Validation	DoH Disabled?	Does it use Public DNS?	DNS Proxy
1	1.1.1.1 + Warp Cloudflare	🟩	Leak not detected	Yes, leaks DNS	🟩	🟩	🟩	🔴	🔴	🔴
2	AirVPN	🟩	Leak not detected	Leak not detected	🔴	🔴	🔴	🟩	🔴	🔴
3	Algo	🟩	Leak not detected	Leak not detected	🟩	🟩	🟩	🟩	🟩	🔴
4	Anonine	🔴	Leak not detected	Leak not detected	🟩	🔴	🔴	🔴	🔴	🔴
5	Astrill VPN	Leaks to ISP	Leaks traffic**	Yes, leaks DNS**	🔴	🟩	🔴	🔴	🟩	🔴
6	Atlas VPN	🟩	Leak not detected	Leak not detected	🟩	🟩	🟩	🔴	🔴	🔴
7	Avast Secureline	🔴	Leak not detected	Leak not detected	🔴	🔴	🔴	🔴	🔴	🔴
8	Avira Phantom	🔴	Leak not detected	Yes, leaks DNS	🟩	🟩	🔴	🔴	🟩	🔴
9	Azire VPN	🟩	Leak not detected	Leak not detected	🟩	🟩	🔴	🔴	🔴	▲
10	BestVPN	🔴	Leak not detected	Leak not detected	🟩	🔴	🟩	🔴	🟩	🔴
11	Betternet	🔴	Leak not detected	Yes, leaks DNS	🔴	🔴	🔴	🔴	🔴	🟩
12	BolehVPN	🔴	Leak not detected	Leak not detected	🟩	🟩	🔴	🔴	🔴	🔴
13	Bullguard	🔴	Leak not detected	Leak not detected	🟩	🟩	🔴	🔴	🔴	🟩
14	Cactus VPN	🔴	Leak not detected	Leak not detected	🟩	🟩	🔴	🔴	🟩	🔴
15	Cryptostorm	🔴	Leak not detected	Leak not detected	🔴	🔴	🔴	🔴	🔴	🔴
16	CyberGhost	🔴	Leak not detected	Leak not detected	🔴	🟩	🟩	🔴	🔴	🔴
17	Encrypt.me	🔴	Leaks traffic**	Yes, leaks DNS**	🟩	🔴	🔴	🔴	🟩	🟩
18	ExpressVPN	🔴	Leak not detected	Leak not detected	🟩	🟩	🔴	🟩	🔴	▲
19	F-Secure Freedome	🟩	Leak not detected	Leak not detected	🟩	🟩	🟩	🔴	🔴	🟩
20	FastestVPN	🔴	Leak not detected	Leak not detected	🟩	🔴	🔴	🔴	🔴	▲
21	Free VPN by Free VPN.org	🔴	Leaks traffic	Yes, leaks DNS	🔴	🔴	🔴	🔴	🟩	🔴
22	Goose VPN	🔴	Leak not detected	Leak not detected	🟩	🟩	🟩	🔴	🔴	🔴
23	Hide My Ass!	🔴	Leaks traffic	Yes, leaks DNS**	🔴	🔴	🟩	🔴	🔴	🔴
24	Hide.me	🟩	Leak not detected	Leak not detected	🟩	🔴	🔴	🟩	🔴	▲
25	HideIPVPN	🔴	Leak not detected	Leak not detected	🟩	🔴	🟩	🔴	🟩	🔴
26	Hotspot Shield	🟩	Leak not detected	Yes, leaks DNS**	🔴	🔴	🟩	🔴	🔴	🟩
27	IP Vanish	🔴	Leaks traffic**	Yes, leaks DNS**	🔴	🔴	🟩	🟩	🔴	🟩
28	IVPN	🟩(in custom)	Leak not detected	Leak not detected	🟩	🔴	🟩	🔴	🔴	🔴
29	Ivacy VPN	🔴	Leaks traffic**	Yes, leaks DNS**	🟩	🔴	🟩	🔴	🔴	🔴
30	K2VPN	🔴	Leak not detected	Leak not detected	🟩	🔴	🔴	🔴	🟩	🔴
31	Kaspersky	🔴	Leak not detected	Leak not detected	▲	🟩	🔴	🔴	🟩	🔴
32	KeepSolid VPN Unlimited	🔴	Leak not detected	Leak not detected	🔴	🔴	🔴	🔴	🔴	🔴
33	LeVPN	🔴	Leak not detected	Leak not detected	🟩	🔴	🟩	🟩	🟩	🔴
34	Mozilla VPN	🟩	Leak not detected	Leak not detected	🟩	🔴	🔴	🔴	🔴	🟩
35	Mullvad VPN	🟩(in custom)	Leak not detected	Leak not detected	🟩	🔴	🟩	🔴	🔴	▲
36	Namecheap	🔴	Leak not detected	Leak not detected	🟩	🔴	🔴	🟩	🔴	🟩
37	NordVPN	🔴	Leak not detected	Leak not detected	🟩	🟩	🔴	🔴	🔴	🔴
38	Norton Secure VPN	Leaks to ISP	Leaks traffic**	Yes, leaks DNS**	🔴	🔴	🔴	🔴	🔴	🔴
39	OVPN	🟩	Leak not detected	Leak not detected	🟩	🟩	🔴	🔴	🔴	🔴
40	OpenVPN Access Server	🔴	Leaks traffic	Yes, leaks DNS	🔴	🔴	🟩	🔴	🔴	🔴
41	Outline	🔴	Leak not detected	Leak not detected	🟩	🟩	🔴	🔴	🟩	🔴
42	Panda VPN	🔴	Leak not detected	Leak not detected	🟩	🔴	🔴	🔴	🟩	🔴
43	Perfect Privacy	🟩	Leak not detected	Leak not detected	🟩	🔴	🔴	🔴	🔴	🔴
44	Private Internet Access	🔴	Leak not detected	Yes, leaks DNS**	🔴	🟩	🟩	🔴	🔴	▲
45	Private Tunnel	🔴	Leak not detected	Leak not detected	🟩	🔴	🔴	🔴	🔴	🟩
46	Private VPN	🔴	Leak not detected	Leak not detected	🔴	🟩	🟩	🔴	🔴	🔴
47	Proton VPN	🔴	Leak not detected	Leak not detected	🟩	🟩	🔴	🟩	🔴	🔴
48	Psiphon	🔴	Leaks traffic	Yes, leaks DNS	🟩	🔴	🟩	🔴	🟩	▲
49	Pure VPN	🔴	Leaks traffic	Leak not detected	🟩	🔴	🔴	🔴	🔴	🔴
50	Riseup	🔴	Leak not detected	Leak not detected	🟩	▲	🟩	🔴	🔴	▲
51	Speedify	🔴	Leaks traffic	Yes, leaks DNS	🟩	🟩	🟩	🔴	🟩	🔴
52	Star VPN	🔴	Leak not detected	Leak not detected	🟩	🔴	🔴	🔴	🟩	🔴
53	Steganos	🔴	Leak not detected	Leak not detected	🟩	🔴	🟩	🔴	🟩	🔴
54	Streisand	🔴	Leak not detected	Yes, leaks DNS	🟩	🟩	🔴	🟩	🟩	🔴
55	Strong VPN	🔴	Leaks traffic**	Yes, leaks DNS**	🟩	🔴	🟩	🟩	🔴	🔴
56	SurfEasy	Leaks to ISP	Leaks traffic	Yes, leaks DNS	🔴	🔴	🟩	🔴	🔴	🔴
57	SurfShark	🔴	Leak not detected	Leak not detected	🟩	🟩	🔴	🔴	🔴	🔴
58	TorGuard	🔴	Leak not detected	Leak not detected	🟩	🔴	🔴	🔴	🔴	🔴
59	Touch VPN	🔴	Leak not detected	Leak not detected	🔴	🟩	🟩	🔴	🟩	🔴
60	Trust.Zone	🔴	Leaks traffic	Yes, leaks DNS	🟩	🔴	🟩	🔴	🟩	🔴
61	TunnelBear	🔴	Leak not detected	Yes, leaks DNS**	🔴	🔴	🟩	🔴	🔴	🟩
62	Turbo VPN	Leaks to ISP	Leaks traffic	Yes, leaks DNS	🟩	🔴	🔴	🔴	🟩	🔴
63	University VPN	Leaks to ISP	Leaks traffic	Yes, leaks DNS	🟩	🔴	🟩	🟩	🔴	🔴
64	Unspyable	🔴	Leak not detected	Leak not detected	🔴	🔴	🟩	🟩	🔴	▲
65	Urban VPN Desktop	🔴	Leaks traffic	Yes, leaks DNS	🔴	🔴	🟩	🔴	🔴	🟩
66	VPN Hotspot - Unlimited Proxy	🔴	Leak not detected	Leak not detected	🟩	🔴	🔴	🔴	🟩	🔴
67	VPN Owl	🔴	Leak not detected	Yes, leaks DNS	🔴	🔴	🔴	🔴	🟩	🔴
68	VPN Plus	🔴	Leak not detected	Leak not detected	🟩	🔴	🔴	🔴	🟩	🔴
69	VPN Pro	🔴	Leak not detected	Leak not detected	🟩	🔴	🟩	🔴	🟩	🔴
70	VPN Proxy Master	🔴	Leaks traffic	Yes, leaks DNS	🟩	🔴	🟩	🔴	🟩	🔴
71	VPN Super: Best VPN Proxy	🔴	Leak not detected	Leak not detected	🟩	🔴	🟩	🔴	🟩	🔴
72	VPN.ac	🔴	Leak not detected	Leak not detected	🟩	🔴	🔴	🔴	🔴	🟩
73	VPNBook	🔴	Leak not detected	Leak not detected	🟩	🔴	🔴	🔴	🔴	🔴
74	VPNLite	🔴	Leak not detected	Leak not detected	🔴	🔴	🔴	🔴	🔴	🔴
75	VPNUK	🔴	Leak not detected	Leak not detected	🔴	🔴	🔴	🔴	🔴	🔴
76	VeePN	🔴	Leak not detected	Leak not detected	🔴	🟩	🔴	🔴	🔴	🔴
77	Vypr	🔴	Leak not detected	Leak not detected	🟩	🔴	🔴	🔴	🔴	🔴
78	Windscribe	🔴	Leak not detected	Leak not detected	🟩	🔴	🟩	🟩	🟩	🔴
79	ZenMate	🔴	Leak not detected	Leak not detected	🔴	🔴	🟩	🔴	🔴	🔴
80	ZoogVPN	🔴	Leak not detected	Leak not detected	🟩	🟩	🔴	🟩	🟩	🔴

Table VI: **VPN Providers & Results**—A red circle indicates "false", and a green square indicates "true" for each test mentioned in the column. A yellow triangle denotes an inconclusive result. Under leaks during tunnel failure, findings also observed in the secure mode (§VI-H) are marked with two asterisks**. For RPKI validation, "true" implies at least one experiment of the provider shows evidence that RPKI validation is enabled.◇