Survey on the Chinese Governments Censorship Mechanisms

Raphael Stadler, Lion Steger*
*Chair of Network Architectures and Services, Department of Informatics
Technical University of Munich, Germany
Email: r.stadler@tum.de, stegerl@net.in.tum.de

***Abstract*—The Chinese government enforces a strict censorship policy on digital content and has created an isolated network for its residents. As it is not feasible to completely disconnect China from the global internet, the government has implemented a powerful and complex system called The Great Firewall of China (GFW) to separate the national network from the rest of the world. This paper focuses on how the Chinese government implements its censorship policies and outlines various techniques the GFW uses to block specific traffic. Subsequently, we present numerous ways on how it is possible to bypass these methods to illustrate the ongoing conflict between the GFW and anti-censorship communities.**

***Index Terms*—Great Firewall of China, GFW, GFC, The Golden Shield Project, DNS poison, SNI filtering, shadowsocks, V2Ray, TOR, Fronting**

1. Introduction

The Great Firewall of China (GFW), also known as The Golden Shield project [1], is a government controlled firewall that acts not only inside the countries network, but also at the interconnections between the national Chinese network and the global networks, the internet. It resides in-between every connection that is initiated to or from China, similar to an attacker like MitM. With this powerful position in the network, the GFW can not only observe and evaluate every connection passively, but it can also act actively by modifying connections or operating maliciously in-between two peers, which is necessary for fulfilling the governments policies. Figure 1 demonstrates the position of the GFW and shows that practically no connection can be initiated without the GFW in the middle.

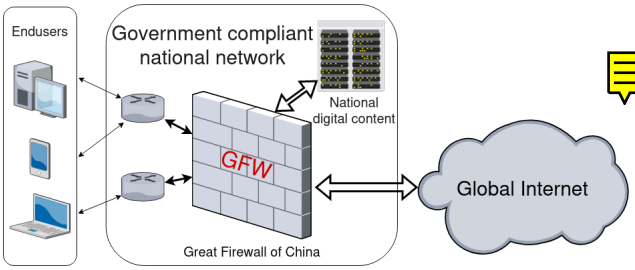

Figure 1: A schematic illustration of the Great Firewall of China and its reach in the Chinese network.

2. Overview of various blocking methods

The GFW has used and is using various means to censor, filter or prevent unwanted traffic. As there are several benefits as well as disadvantages on each method presented below, the practical implementation of the GFW adjusts its sensors dynamically in complex and obscure manners. The requirement for this is driven by the ongoing evolution of technology and the potential for changes in bypassing methods.

2.1. Subnet Blocking and Re-routing

One of the simplest methods the GFW is using is blocking whole IP subnets [2]. This can be done by modifying routing tables and re-routing specific IP subnets. These altered routes either redirect traffic to GFW controlled servers trying to mimic the actual target service, and therefore may gain data for further analysis, or they can be null routed, which means that effectively any connection made to the destination is prevented. Alternatively, the GFW can accomplish similar results by using BGP hijacking. BGP hijacking works by maliciously announcing unowned and improper BGP prefixes. While these altered BGP prefixes are normally announced in the national networks only, it does happen that announcements also get accepted by international networks as well, if they are announced incorrectly or with harmful intents. [3]

Both methods can lead to collateral damage, as IP addresses might be shared by various services, which are then blocked as well. IP blocks or BGP hijacking are commonly used by the GFW, as they allow to block a service quickly and effectively. These blocks are often based on research, such as connection analysis or publicly available information. The GFW blocks have cause great collateral damage [4], [5], which is another reason why the GFW implements a fairly dynamical concept and lifted various blocks again.

2.2. DNS Poisoning

Another method of blocking or redirecting traffic is *DNS Poisoning*. When using a DNS resolver inside of the Chinese network, it must obey the law and enforce the governments policies. A DNS request that contains a blocked hostname resolves to a different IP address than internationally announced. [6] The GFW uses a more sophisticated approach to censor internet access for resolvers that are outside of the Chinese network. It lets resolve the request correctly and leaves it untouched

in the outgoing direction and sends a crafted response back to the client that initiated the connection. The GFW effectively spoofs the DNS resolver's IP and exchanges the payload. This is possible as there is no authentication or encryption in the DNS protocol. [7] While possible to filter the actual response, the GFW allows, to some extent, the original response to pass through, which leads to race conditions on which DNS reply is faster. One example of DNS poisoning happened in 2013, when the website *github.com* was blocked in China. [4] Later it was lifted again due to an unforeseen amount of protest in the Chinese community. Another example where DNS packets were inspected occurred in 2002 when *web.mit.edu* was filtered due to resolving to the same IP address as another hostname *www.falundafa.org*, which had been banned by the Chinese government. [5]

2.3. Encrypted DNS

DNS Poisoning can be bypassed by using encrypted DNS requests with implementations such as DNS-over-HTTPS (DoH), DNS-over-TLS (DoT), DNS-over-QUIC (DoQ) or DNSCrypt. These protocols require that the resolver remains outside of the censored network to work as intended. [7] The encrypted nature of some network requests makes it difficult for the GFW to analyse or modify the packets. The GFW has shown that it simply can block encrypted DNS requests, which makes these protocols only partially effective as bypassing strategies [8]. Encrypted DNS requests are not a reliable solution to bypass DNS poisoning, if the resolver is controlled by the GFW. In fact, using a DoH resolver operated by Alibaba can yield similar incorrect responses as using unencrypted Chinese resolvers. Even though the encryption has not been broken, it suggests that the resolvers operated by Alibaba are at least working in cooperation with the GFW. [8]

2.4. Keyword Filtering

The GFW is able to detect keywords in network requests by inspecting traffic and terminates connections accordingly. [9] This process is similar to DNS poisoning, where an unwanted term can be identified in a plain and unencrypted request. While not limited to the TCP protocol, a good indicator for this mechanism can be observed in TCP-based traffic, as the GFW reacts by sending a *TCP-RST* packet back on the same channel, effectively terminating any further communication between two peers. [10] This method of identification and reaction from the GFW is practically possible on every protocol transmitting plaintext, such as SMTP, IMAP, POP or TELNET. Due to the prevalence of HTTP-based traffic, the GFW has developed specialized capabilities for detecting keywords in HTTP-based traffic. [11] When a user attempts to access a website, the GFW's keyword filtering system inspects the URL or the content of the page to see if it contains any keywords that are on a pre-defined list of prohibited words or phrases. If the website or page contains prohibited keywords, it will block the request and prevent the user from accessing the website. The GFW is especially enforcing this technique in Chinese search engines, such as *google.cn* or *baidu.cn* [12]

2.5. SNI-Filtering

With an increase of implementing TLS on web-servers [2] and using encrypted connections, this approach is ineffective if the corresponding target server does not comply to the Chinese governments policies by letting the GFW inspect the unencrypted content. While companies inside of China often cooperate with the government [13], this is mostly the case for services operating outside of China. As IP addresses are often shared, a technique called *Server Name Indication* (SNI) is currently widely implemented to serve encrypted content for multiple hostnames. SNI helps in identifying which specific service is to be provided by sending the requested hostname in plaintext, as it is required and sent during the unencrypted connection setup. This allows the GFW to specifically filter connections based on the hostname, while minimizing collateral damage and allowing other hostnames to continue using the IP address. The detection rate on such connections is limited to hostname filtering. This vulnerability of SNI can be remedied by using encrypted SNI. ESNI works by first retrieving a trusted encryption key and sending the SNI header encrypted to the web-server. It was further investigated, that the GFW does simply filter ESNI connections currently, by purely dropping detected packets. [2] It blocks only a certain ESNI version, while leaving other protocol implementations with other versions untouched. [14]

2.6. Other detection methods

Similarly to ESNI, the GFW has difficulties detecting and verifying content when the *QUIC* protocol is used, due to the implicit encryption. Currently, blocks are less stringent and QUIC requests are mostly passing through. It is expected that QUIC packets might be dropped more aggressively or that the GFW maintainers implement inspection algorithms, if possible. [15]

Another method the GFW implements is bandwidth throttling. While this is not a practical way of blocking connections, it discomforts the use of bypassing methods for end-users, which might achieve the effect that they give up or reduce their use of bypassing methods. Especially for international connections, throttling is generally in place. [16]

While not commonly linked to casual internet connections, using torrents allows accessing and downloading files as well. It was shown, that one of the biggest Chinese ISPs, namely China Telecom, does prevent torrenting, likely in compliance to the Chinese governments strategy. [17]

2.7. Combining methods

The GFW is capable to implement multiple methods to better enforce their blocking strategy. For example, while bypassing DNS poisoning and receiving the correct IP address when using some DNS bypassing technique, the GFW can still block connections to the target, if HTTP would be used. A scenario was shown, where even using connections based on HTTPS with enabled SNI would be blocked solely by the leaked hostname, despite bypassing the DNS poisoning of the GFW. [8]

3. General bypassing methods

One great drawback of previously discussed bypassing methods is the demand of specific implementations on the content delivering server. As an end-user normally has no control over such single services, it is beneficial for them to rely on some form of general circumvention technique for explicitly selected or all connections. The systems presented below work by sending encapsulated traffic to a middle server, which then unwraps the original request and executes it in position for the end-user.

3.1. Bypassing via proxies

In general a technical server proxy is a piece of software, that acts as a gateway between two machines. Proxies can act on different layers in the OSI/ISO motel, the current common services however mostly act on the highest layers, for instance using HTTP and HTTPS (Layer 7) or the SOCKS protocol (Layer 5). Some common software that is used for proxies are shadowsocks, obfs (by TOR), Trojan or V2Ray. [18]–[20] A proxy, by design, needs to be configured for the specific software that should use it. Meaning, that it normally is only used by an explicit application which then encapsulates it's requests into the proxy specific protocol.

3.2. Bypassing via VPNs

Another common method is to use virtual private networks (VPNs). Despite that the general idea of a VPN is not meant for utilizing it to bypass restrictive censorship, it is a practical utility for doing so. [21] VPNs, while very similar to proxies, normally act on lower layers, by using implementations such as OpenVPN, WireGuard or IPSec. They therefore have more overhead in comparison to proxies but can route traffic of all applications through the tunnel. This is specifically useful for application that do not support proxies. [22]

3.3. Other bypassing methods

Another viable option for end-users to use is a *Web Proxy*. [23], [24] While quite similar to a regular proxy in terms of functionality, a web proxy is an application hosted on a web-server, that is only designed to allow users to browse websites via this specific server. [25]

Using international SIM cards is another possible solution to bypass censorship, as this specific traffic is routed to the mobile network service provider and therefore normally not filtered at all. To reduce latency, SIM cards from ISPs in Hong Kong are currently the most favoured by the Chinese communities.

4. Detection and verification of circumvention services

Significant efforts have been made to bypass the GFW, but many systems encounter challenges in the identification of connections based solely on recognizable properties. Passive detection of connections relies on specific identifiers, such as port and protocol. When using standard software configurations, fixed ports are often used, enabling the easy identification of connections. By default, OpenVPN is listening on UDP port 1194 and IPSec uses UDP and TCP on specific ports and the rather ESP protocol. As a result, it is straightforward to detect and block such connections for the GFW. [26], [27] Additionally, it is possible to further detect specific VPN software implementations based on a technique called *Fingerprinting*.

4.1. OpenVPN

Especially as OpenVPN is currently widely used in commercial VPN software, it is a high priority target for the GFW maintainers. [28] While some identifiers are based on publicly available data, such as WHOIS information, AS numbers of VPN providers or PTR records of their IP addresses, it was shown that it is possible to detect OpenVPN traffic based on properties such as the unencrypted operational byte patterns in encapsulated packets. Although this detection technique already has a high accuracy of above 99% while only needing ten connection packets, it is also possible to recognize TCP based OpenVPN traffic on other properties, such as specific TCP options. [29] Although there are some proposed protocol changes that would obfuscate OpenVPN-based traffic, some are disregarded by the OpenVPN developers, such as an XOR scrambler. Others often rely on different obfuscation protocols such as shadowsocks, obfs or even proprietary protocols, all with their own disadvantages. [28] OpenVPN is currently not useful in bypassing very strictly censored networks, such as the national networks in China.

4.2. TOR and obfs

Another detection method uses fingerprinting techniques for TOR-based traffic. Initially, the GFW blocked TOR by using publicly available information on TOR entry servers. The TOR community responded by introducing unpublished TOR bridges, but the GFW was still able to detect these bridges by identifying the used ciphers. Trying to mitigate this by using obfuscators, it was still possible for the GFW to detect bridges by using *Active Probing*. Active probing involves scanning for keywords on third-party service providers such as websites, forums, or similar platforms, and analysing VPN or proxy services for the purpose of blocking them. In the case of TOR-based traffic, if the GFW detected a connection with a likely chance of being used in TOR related traffic, it then would send specifically crafted packets to the suspected TOR bridge and confirm and block the IP address, if it was responding as expected. It was shown that the GFW could block an unpublished TOR bridge practically instantly after the first connection was initiated from an end-user in China. [30]

4.3. Shadowsocks

Shadowsocks-based proxies with early implementations were found to be susceptible to detection due to a lack of authentication. [19] By sending specific packet

headers to a proposed server, an attacker could exploit, that a proxy would respond. [22] With this response, the GFW could then verify that it was a proxy service and therefore block connections to the IP address. This attack could partially be mitigated by implementing an OTA (One-Time Auth) mechanism on the payload. A later improved protocol specification using AEAD ciphers effectively addressed this vulnerability in shadowsocks. [31]

5. Mimicking and Fronting

To bypass the GFW, VPN or proxy service operators often try to mimic commonly used services and therefore reduce the risk of detection. HTTPS is an excellent protocol for such concealment, as very much of the traffic on the internet is HTTP/HTTPS based. [11] Additionally, HTTPS traffic is already encrypted, which allows to easily reuse the protocols properties for building encrypted tunnels. This is also one reason, why many tunnels specifically designed for bypassing strict censorship do mimic HTTPS connections, currently with a notable success rate.

To further prevent blocks, bypassers often use *Fronting* to increase collateral damage if blocks are actually enforced. When implementing it, another provider is put in front of the actually bypassing service. For instance, when setting up a private VPN service, it helps to host such service on popular cloud service providers, such as Amazon Web Services. With those in place, the GFW could not ban the whole network, as legitimate services are operating on the same platforms as well. Single IPs can however still be blocked.

5.1. Importance of CDNs

Therefore, it is a better method to use CDNs as a fronting mechanism. As much of the HTTPS traffic is cached and accelerated using commercial Content Delivery Networks, they can also be used in fronting for tunnel services as well, if the tunnel uses HTTPS. For instance, at the time of this research, one of the most recent tunnelling setups would not only include the combination of a CDN and a HTTPS (forward) proxy, namely V2Ray, but also a HTTPS reverse proxy (nginx) to allow custom webserver content to be hosted on the same machine as well. [32] [33] This reverse proxy helps in mitigating blocks, as there might be actual content displayed if visiting the website normally and therefore could look like a normal webpage. To the GFW, it is therefore much harder to block the connection, as it appears to be regular HTTPS traffic, and because the IP address of the actual proxy is hidden behind the CDN's IP address. [34] [35]

These benefits apply to web proxies as well, as they have the great advantage of being used in combination with highly advanced fronting methods very easily. Despite the limitations, it might remain one of the most undetected proxying methods, as the application does not look different to normal content hosted on web-servers and further, as there are a lot different implementations from different creators as well.

The GFW's current tries to prevent these setups is done by throttling international CDNs and forcing companies to setup their infrastructure in China as well. [16] These servers in China must obey the law, which allows the government to access data or enforce their own policies on these companies relatively easily. Companies, such as *Apple Inc.*, censor content even without the Chinese government interfering officially. [13]

6. Verification of GFW's blocking techniques

In the course of this project, some experiments to test the GFW's capabilities from outside of the Chinese censored networks have been setup. This works, because the GFW often acts in a symmetric way.

We could prove the existence of DNS poisoning by sending DNS queries to Chinese based IP addresses. Trying to resolve *google.com*'s AAAA record via IPv6 connections returned invalid IPv6 addresses. It should be noted that the GFW intercepts packets before they reach their intended destination, eliminating the need for the targeted IP addresses to provide DNS resolution services.

Another blocking technique we tried to verify was keyword filtering and SNI-Filtering. For this experiment, we made use of the command-line tool *curl* to actively resolve a hostname to a Chinese IP for a HTTP request. When connecting to the valid Chinese IP, such as the IP *140.205.174.2* that belongs to *alibaba.cn*, we could experience connection resets when we tried to connect to it by using various known blocked hosts, such as *facebook.com* or *wikipedia.org*. When using not blocked hosts, such as *tencent.cn* or *google.cn* the connection was established and content was transmitted.

Furthermore, even for HTTPS requests a similar result could be achieved. For *facebook.com*, *wikipedia.org* and others, the TLS handshake was interrupted due to connection resets. When using invalid hostnames, such as *baidu.cn* or *tencent.cn*, a TLS certificate mismatch could be conducted. This expected result suggests that the connection was not interrupted and the TLS handshake completed as expected. When connecting with the actual hostname *alibaba.cn* content was transmitted as expected.

7. Conclusion and future work

The Great Firewall of China is a very powerful tool in enforcing the strict censorship policies the Chinese government sets up. It is a strong counterpart to the freedom of speech communities by demonstrating the possibilities and techniques it can implement and execute in the Chinese national network. However, the GFW is not perfect and there is much work done to bypass it, not only from the inside of China but also internationally. Especially when combining multiple bypassing methods and with the development of new designs and protocols, the Chinese government needs to perfectly balance between isolating its citizens from global information and separate itself too much from the rest of the world.

References

[1] S. Chandel, Z. Jingji, Y. Yunnan, S. Jingyao, and Z. Zhipeng, "The Golden Shield Project of China: A Decade Later—An in-Depth Study of the Great Firewall," in *2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, Oct. 2019, pp. 111–119.

[2] Z. Chai, A. Ghafari, and A. Houmansadr, "On the Importance of Encrypted-SNI (ESNI) to Censorship Circumvention," p. 8.

[3] C. Demchak and Y. Shavitt, "China's Maxim – Leave No Access Point Unexploited: The Hidden Story of China Telecom's BGP Hijacking," *Military Cyber Affairs*, vol. 3, no. 1, Jun. 2018, accessed on 2022-11-21. [Online]. Available: http://scholarcommons.usf.edu/mca/vol3/iss1/7/

[4] S. N. Company, "Sudo Null - Latest IT News," accessed on 2022-11-21. [Online]. Available: https://sudonull.com/post/132004-Github-is-blocked-in-China

[5] "China Blocks MIT Web Addresses - The Tech," accessed on 2022-11-21. [Online]. Available: http://tech.mit.edu/V122/N58/58web.58n.html

[6] O. Farnan, A. Darer, and J. Wright, "Poisoning the Well: Exploring the Great Firewall's Poisoned DNS Responses," in *Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society*. Vienna Austria: ACM, Oct. 2016, pp. 95–98, accessed on 2022-11-21. [Online]. Available: https://dl.acm.org/doi/10.1145/2994620.2994636

[7] C. Kwan, P. Janiszewski, S. Qiu, C. Wang, and C. Bocovich, "Exploring Simple Detection Techniques for DNS-over-HTTPS Tunnels," in *Proceedings of the ACM SIGCOMM 2021 Workshop on Free and Open Communications on the Internet*. Virtual Event USA: ACM, Aug. 2021, pp. 37–42, accessed on 2022-11-21. [Online]. Available: https://dl.acm.org/doi/10.1145/3473604.3474563

[8] L. Jin, S. Hao, H. Wang, and C. Cotton, "Understanding the Impact of Encrypted DNS on Internet Censorship," in *Proceedings of the Web Conference 2021*. Ljubljana Slovenia: ACM, Apr. 2021, pp. 484–495, accessed on 2022-11-21. [Online]. Available: https://dl.acm.org/doi/10.1145/3442381.3450084

[9] Z. Weinberg, D. Barradas, and N. Christin, "Chinese Wall or Swiss Cheese? Keyword filtering in the Great Firewall of China," in *Proceedings of the Web Conference 2021*, ser. WWW '21. New York, NY, USA: Association for Computing Machinery, Jun. 2021, pp. 472–483, accessed on 2022-11-30. [Online]. Available: https://doi.org/10.1145/3442381.3450076

[10] J. Zittrain and B. Edelman, "Internet filtering in China," *IEEE Internet Computing*, vol. 7, no. 2, pp. 70–77, Mar. 2003, conference Name: IEEE Internet Computing.

[11] R. Fontugne, P. Abry, K. Fukuda, D. Veitch, K. Cho, P. Borgnat, and H. Wendt, "Scaling in Internet Traffic: A 14 Year and 3 Day Longitudinal Study, With Multiscale Analyses and Random Projections," *IEEE/ACM Transactions on Networking*, vol. 25, no. 4, pp. 2152–2165, Aug. 2017, accessed on 2022-11-24. [Online]. Available: http://ieeexplore.ieee.org/document/7878657/

[12] J. S. O'Rourke, B. Harris, and A. Ogilvy, "Google in China: government censorship and corporate reputation," *Journal of Business Strategy*, vol. 28, no. 3, pp. 12–22, May 2007, accessed on 2022-11-21. [Online]. Available: https://www.emerald.com/insight/content/doi/10.1108/02756660710746229/full/html

[13] S. Liao, "Apple officially moves its Chinese iCloud operations and encryption keys to China," Feb. 2018, accessed on 2022-12-01. [Online]. Available: https://www.theverge.com/2018/2/28/17055088/apple-chinese-icloud-accounts-government-privacy-speed

[14] "Exposing and Circumventing China's Censorship of ESNI," Aug. 2020, accessed on 2022-11-21. [Online]. Available: https://geneva.cs.umd.edu/posts/china-censors-esni/esni/

[15] K. Elmenhorst, B. Schütz, N. Aschenbruck, and S. Basso, "Web censorship measurements of HTTP/3 over QUIC," in *Proceedings of the 21st ACM Internet Measurement Conference*. Virtual Event: ACM, Nov. 2021, pp. 276–282, accessed on 2022-12-01. [Online]. Available: https://dl.acm.org/doi/10.1145/3487552.3487836

[16] P. Zhu, K. Man, Z. Wang, Z. Qian, R. Ensafi, J. A. Halderman, and H. Duan, "Characterizing Transnational Internet Performance and the Great Bottleneck of China," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 4, no. 1, pp. 1–23, May 2020, accessed on 2022-12-01. [Online]. Available: https://dl.acm.org/doi/10.1145/3379479

[17] M. Dischinger, A. Mislove, A. Haeberlen, and K. P. Gummadi, "Detecting bittorrent blocking," in *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement conference - IMC '08*. Vouliagmeni, Greece: ACM Press, 2008, p. 3, accessed on 2022-11-21. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1452520.1452523

[18] P. Liubinskii, "The Great Firewall's active probing circumvention technique with port knocking and SDN," p. 65.

[19] X. Zeng, X. Chen, G. Shao, T. He, Z. Han, Y. Wen, and Q. Wang, "Flow Context and Host Behavior Based Shadowsocks's Traffic Identification," *IEEE Access*, vol. 7, pp. 41 017–41 032, 2019, conference Name: IEEE Access.

[20] Z. Deng, Z. Liu, Z. Chen, and Y. Guo, "The Random Forest Based Detection of Shadowsock's Traffic," in *2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, vol. 2, Aug. 2017, pp. 75–78.

[21] P. Ferguson and G. Huston, "What is a VPN - Univ-pau.fr," Apr. 1998, accessed on 2022-12-01. [Online]. Available: https://cpham.perso.univ-pau.fr/ENSEIGNEMENT/COMMUN/vpn_ferguson.pdf

[22] Alice, Bob, Carol, J. Beznazwy, and A. Houmansadr, "How China Detects and Blocks Shadowsocks," in *Proceedings of the ACM Internet Measurement Conference*. Virtual Event USA: ACM, Oct. 2020, pp. 111–124, accessed on 2022-11-24. [Online]. Available: https://dl.acm.org/doi/10.1145/3419394.3423644

[23] "PHP Online Web Proxy," Nov. 2022, accessed on 2022-11-24. [Online]. Available: https://github.com/NicheOffice/php-web-proxy

[24] Athlon1600, "php-proxy-app," Nov. 2022, accessed on 2022-11-24. [Online]. Available: https://github.com/Athlon1600/php-proxy-app

[25] J. Dick, "A simple PHP web proxy." Nov. 2022, accessed on 2022-11-24. [Online]. Available: https://github.com/joshdick/miniProxy

[26] A. Cain and A. Proctor, "OpenVPN Access Server System Administrator Guide," p. 58.

[27] K. Seo and S. Kent, "Security Architecture for the Internet Protocol," Internet Engineering Task Force, Request for Comments RFC 4301, Dec. 2005, accessed on 2022-12-01. [Online]. Available: https://datatracker.ietf.org/doc/rfc4301

[28] D. Xue, R. Ramesh, A. Jain, M. Kallitsis, J. A. Halderman, J. R. Crandall, and R. Ensafi, "OpenVPN is Open to VPN Fingerprinting," p. 19.

[29] Y. Pang, S. Jin, S. Li, J. Li, and H. Ren, "OpenVPN Traffic Identification Using Traffic Fingerprints and Statistical Characteristics," in *Trustworthy Computing and Services*, Y. Yuan, X. Wu, and Y. Lu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 443–449.

[30] A. Dunna, C. O'Brien, and P. Gill, "Analyzing China's Blocking of Unpublished Tor Bridges."

[31] "Shadowsocks active-probing attacks and defenses," accessed on 2022-11-21. [Online]. Available: https://groups.google.com/g/traffic-obf/c/CWO0peBJLGc/m/Py-clLSTBwAJ

[32] "V2Ray (WebSocket + TLS + Web + Cloudflare) - Eric," accessed on 2022-11-21. [Online]. Available: https://ericclose.github.io/V2Ray-TLS-WebSocket-Nginx-with-Cloudflare.html

[33] "Use Cloudflare Certificate with Nginx + V2Ray + Websocket +TLS + CDN," Aug. 2020, accessed on 2022-11-21. [Online]. Available: https://henrywithu.com/use-cloudflare-certificate-with-nginx-v2ray-websocket-tls-cdn/

[34] S. Satija and R. Chatterjee, "BlindTLS: Circumventing TLS-based HTTPS censorship," in *Proceedings of the ACM SIGCOMM 2021 Workshop on Free and Open Communications on the Internet*. Virtual Event USA: ACM, Aug. 2021, pp. 43–49, accessed on 2022-11-21. [Online]. Available: https://dl.acm.org/doi/10.1145/3473604.3474564

[35] D. Fifield, C. Lan, R. Hynes, P. Wegmann, and V. Paxson, "Blocking-resistant communication through domain fronting," *Proceedings on Privacy Enhancing Technologies*, vol. 2015, no. 2, pp. 46–64, Jun. 2015, accessed on 2022-11-21. [Online]. Available: https://petsymposium.org/popets/2015/popets-2015-0009.php