Internet Censorship Circumvention Protocols
Are Shadowsocks and Trojan-go Still Relevant?

By nthLink Engineering Team

November 2023

The evolving landscape of internet censorship, particularly under authoritarian regimes, demands increasingly sophisticated methods for secure and unrestricted online communication. This whitepaper offers in-depth analysis of various censorship circumvention tools and techniques, with a focus on Shadowsocks and Trojan-go (also commonly known as “Trojan”) technologies.

Notwithstanding their utility, Shadowsocks and Trojan-go are vulnerable to detection and blocking by the Great Firewall of China (GFW)—a combination of legislation and technology enforced by the People’s Republic of China to regulate the internet domestically. The GFW has upped its game through real-time blocking and deep packet inspection. We’ll explore Shadowsocks and Trojan-go, along with alternative technologies for bypassing censorship, including VPNs, Tor, and emerging tools like nthLink that employ multiple protocols (including Trojan-go and V2Ray).

The overarching argument is that no single protocol can claim supremacy; instead, the effectiveness of a circumvention method is contingent on various factors including geography, service provider, and even the specific route data travels within a country. This whitepaper aims to guide users and developers alike in choosing or designing the most appropriate tools for defeating censorship, rooted in the principle that the best protocol is that which can be tailored to individual needs.

Internet Censorship

Authoritarian governments employ a multifaceted approach to censor the internet, using technical, legal, and social methods to control information flow and suppress dissent. Filtering and blocking systems are commonly used to restrict access to specific websites and online services. Legal measures criminalize certain content and force compliance from internet service providers and platforms.

Surveillance is another significant aspect, with authorities monitoring citizens' online activities. Social tactics involve propaganda, state-controlled media, and manipulation of online discussions. Additional tactics like throttling (limiting internet speed), internet shutdowns, and establishing national internet infrastructures are also employed, while collaboration with technology companies can subtly facilitate censorship.

An Overview of Various Censorship Circumvention Tools and Techniques

Internet users in authoritarian countries use various techniques to bypass internet censorship

and access blocked content. Virtual Private Networks (VPNs) create encrypted connections, while proxy servers like Shadowsocks and Tor act as intermediaries to mask users' identities. Users may also resort to alternative DNS servers, steganography (concealing messages or information within other nonsecret text or data), and offline methods to share information. Additionally, encrypted messaging apps, domain fronting, and collaborative efforts to maintain open internet access contribute to circumvention efforts. By adopting these methods, users can communicate more freely and maintain privacy despite government-imposed restrictions.

This paper will focus specifically on the following circumvention technologies:

Shadowsocks

Shadowsocks is an open-source SOCKS5 proxy designed to help users bypass internet censorship by encrypting and redirecting internet traffic through a remote server. A SOCKS5 proxy is the fifth version of the SOCKS (Socket Secure) internet protocol, used to route network traffic between a client and server through a proxy server. SOCKS5 offers features like various authentication methods, IPv6 support, and the ability to handle both Transmission Control Protocol and User Datagram Protocol traffic. It's commonly used for tasks like web scraping, bypassing geo-restrictions, and enhancing online security and privacy. Overall, it provides a reliable and user-friendly solution for internet users seeking unrestricted access in authoritarian environments.

Trojan-go

Trojan-go is an open-source proxy software designed to circumvent internet censorship while mimicking regular HTTPS traffic. The primary objective is to blend in with normal web traffic to avoid detection and blocking by firewalls or other network filters. When a client wants to visit a blocked site, it first connects to a Trojan-go server. The client and server exchange data using a Trojan-go protocol wrapped in TLS encryption, making it look like standard HTTPS traffic.

V2Ray

V2Ray is an open-source network proxy tool to help bypass internet censorship and improve security. It's designed to be versatile, supporting multiple protocols, including its custom protocol called VMess. V2Ray can run on various platforms and be configured to utilize different obfuscation techniques, making it more difficult for third parties to detect and block your connection.

The VMess protocol is particularly noteworthy for its security features. It's designed to be both secure and fast, using a combination of multiple encryption methods and dynamic session keys to secure user data. Each client-server pair in a VMess connection has its own randomly generated session keys, making it difficult for eavesdroppers to decrypt the data stream. VMess also includes measures to protect against replay attacks, ensuring a secure and reliable connection. A replay attack occurs when a malicious actor eavesdrops on a secure network communication, intercepts it, and then delays it or resends it to misdirect the receiver into doing what the malicious actor wants.

Transport Layer Security in Transport Layer Security (TLS-in-TLS)

The TLS-in-TLS concept originates from the widespread use of HTTPS, a secure version of HTTP. HTTPS employs Transport Layer Security (TLS) to encrypt plaintext HTTP data, making it safe for internet transmission. When using proxy tools like Trojan-go or vmess+tls, another layer of TLS encryption is added before sending the data to the node server, resulting in a characteristic nested encryption scenario typical of TLS-in-TLS.

Shadowsocks: Limitations, Blocking, and Alternatives

Limitations

Shadowsocks offers flexibility with various encryption methods, widespread platform compatibility, and ease of use. Though it has limitations, including not being designed for complete anonymity, lacking centralized management tools, and potentially resulting in slower connection speeds.

It’s primarily designed for bypassing internet censorship rather than ensuring complete anonymity. While it encrypts data between the client and the server, it does not route traffic through multiple nodes or implement additional privacy features like data obfuscation. As a result, it may not sufficiently hide user activity from all types of monitoring or surveillance, making it less suitable for those seeking full anonymity.

Originally designed for individual use rather than enterprise-level applications, Shadowsocks lacks the functionalities required for large-scale deployments. Consequently, orchestrating and managing a network of Shadowsocks servers to accommodate tens of thousands of users presents a considerable challenge. The system's reliance on shared-key encryption, compounded by the absence of a centralized key management solution, renders load-balancing across a server farm both complicated and inefficient. This can lead to performance bottlenecks, where some servers may become overburdened and sluggish, while others remain underutilized.

Blocking

The GFW uses sophisticated techniques to detect and block Shadowsocks, despite its custom protocol and encryption. These methods include deep packet inspection (DPI) to analyze traffic patterns, active probing to identify servers running Shadowsocks, and modified probing to detect Shadowsocks disguised as HTTPS websites. (*While both "active probing" and "modified probing" involve sending packets to probe servers, active probing is a general technique, and modified probing is a specialized form aimed at defeating attempts to disguise the Shadowsocks server.*) The GFW also monitors and blocks IP addresses of known Shadowsocks servers and hosting providers frequently used for Shadowsocks deployment. Additionally, the GFW employs machine learning algorithms, manipulates DNS requests, and uses manual investigation to maintain control over information flow and block Shadowsocks servers.

As of November 6, 2021, the GFW has initiated real-time blocking of Shadowsocks and other fully encrypted proxies. Consequently, any new Shadowsocks servers identified by the GFW will be automatically blocked. This indicates an evolution in China's internet censorship capabilities, moving beyond passive traffic analysis to dynamic real-time blocking. The GFW's new system impacts various mainstream circumvention tools, affecting their functionality partially or completely. Researchers from GFW Report have investigated this censorship system and its impact on popular circumvention tools like Shadowsocks, Outline, VMess, and others.

Despite the inherent limitations and ability of the GFW to detect it, Shadowsocks continues to be a popular and effective tool for circumventing internet restrictions. Users have adopted a range of innovative strategies to counteract evolving censorship tactics. These strategies encompass server rotation for periodic IP address changes, the use of dynamic ports to continually alter communication channels and elude port-based blocking, as well as selective routing to direct only sensitive or blocked traffic through Shadowsocks servers.

Additionally, traffic padding is employed to insert random data into packets, thereby complicating the task of deep packet inspection for identification. Recently, researchers at Google Jigsaw have taken this a step further by introducing a "prefix" feature in their Outline app, a VPN based on Shadowsocks, to enhance its capability to evade detection by the GFW.

Alternatives

Multiple alternative technologies to Shadowsocks exist for circumventing internet censorship and accessing restricted content. VPNs create encrypted connections to remote servers, masking IP addresses and bypassing censorship. The Tor network enhances anonymity but may be slower and blocked in some countries. Proxy servers, like HTTP/S proxies, effectively bypass certain censorship methods. Alternative DNS servers and circumvention tools like nthLink, Psiphon, and Lantern combine various techniques for secure internet access.

nthLink supports multiple protocols, such as Trojan-go, V2Ray, and Shadowsocks—offering users tailored solutions based on their specific censorship circumstances. Trojan-go excels at disguising internet traffic as standard HTTPS, thereby thwarting DPI. In contrast, V2Ray enhances both security and speed through its proprietary VMess protocol.

nthLink's streamlined user interface effectively shields users from the complex task of selecting and configuring the right protocols, making it easier for people to navigate the often complex configurations associated with Trojan-go, V2Ray, Shadowsocks, and others, and to maintain online privacy. Catering to non-technical users, nthLink automatically selects the most suitable protocol and connection parameters based on each user's device and geographical location.

The “Trojan Killer”

In an era where secure and reliable data transmission is non-negotiable, nthLink has made a significant impact by adopting the Trojan-go protocol as its standard method of data transportation. This calculated move aimed to enhance speed, scalability, and stability while

also evading detection. It's a strategy that has largely paid off, as evidenced by an overwhelming wave of positive user feedback. However, it's essential to note that even with this success, Trojan-go remains susceptible to detection and possible blocking—a concern highlighted in discussions dating back to 2017.

Initially, when Trojan-go debuted in 2017, the consensus was that firewalls lacked the capability to detect TLS-in-TLS. This belief held ground for an extended period, thanks in part to the robust connectivity offered by Trojan-go and vmess+tls nodes. However, the status quo was disrupted around October 3, 2022, when more than 100 users in the circumvention community reported blocking of TLS-based circumvention protocols such as Trojan-go, Xray, V2Ray TLS+Websocket, VLESS, and gRPC. This cast a spotlight on the TLS-in-TLS issue once again.

While the mechanics of the GFW remain largely a mystery, anecdotal user feedback suggests that some form of TLS-in-TLS detection mechanism may exist. However, it's important to clarify that even if TLS-in-TLS detection exists, it doesn't automatically mean all TLS-in-TLS-enabled nodes will be blocked. The efficacy of this detection may vary depending on factors like the internet service provider and geographic location. In some regions, restrictions are less stringent, allowing even easily detectable protocols like Shadowsocks to remain operational despite the capabilities of the GFW.

The ability to detect TLS-in-TLS by the GFW is theoretically straightforward but intricate in practice. One reason for this complexity is the impact of nested encryption on data size. While TLS-in-TLS increases data size and differentiates it from standard HTTPS traffic, that increase alone is not always a reliable indicator for detection due to various other factors that might affect data size.

Another complicating factor is the temporal characteristic involved in TLS-in-TLS detection. Standard HTTPS traffic commences directly following a single TLS handshake. However, TLS-in-TLS incorporates an additional, internal TLS handshake within the encrypted data of the outer TLS layer. This extra handshake process could vary in its timing, making it more challenging to identify definitively based on temporal patterns alone.

In May 2023, the landscape of TLS-in-TLS detection underwent a significant shift following the release of a proof-of-concept program by Xray's creator, RPRX. Dubbed "Trojan-Killer," the program upended the prevailing belief that detecting TLS-in-TLS required immense computational resources. It revealed that Trojan-go servers could be accurately pinpointed by simply analyzing the size of data packets during the handshake phase, challenging previous assumptions about the computational complexity of TLS-in-TLS detection.

V2Ray’s “Vision” Feature: A Response to Counter TLS-in-TLS Detection

In response to the GFW's capabilities for TLS-in-TLS detection, V2Ray introduced a new

feature known as XTLS and later rolled out a refined version called “Vision” (xtls-rprx-vision). The primary aim of Vision is to manage data streams and inject additional data into the TLS handshake, thereby altering the characteristic TLS-in-TLS features that could potentially be flagged by the GFW.

However, there's an important caveat to consider: enabling Vision in V2Ray comes at a cost. Specifically, service providers lose the ability to employ Content Delivery Networks (CDNs), which leaves the V2Ray server's IP address vulnerable to exposure and subsequent blocking.

This overview aims to delve into the intriguing complexities of Trojan-go and TLS-in-TLS, exploring potential vulnerabilities and pondering preventative strategies and solutions. With the landscape of censorship technologies constantly evolving, understanding these dynamics is crucial for both users and service providers.

Conclusion

The supremacy of a protocol is subjective; the best protocol is the one tailored to individual needs. This is one of the main reasons that nthLink adopted the multi-protocol architecture that allows clients to switch communication methods on-the-fly. The continued viability of Shadowsocks, Trojan-go, and other protocols, or the necessity of setting up a node capable of addressing the TLS-in-TLS feature remains to be validated through continued experimentation.

Factors such as geographical region, service provider, VPS vendor, datacenter, protocol, and in-country route that the data travels through could all potentially influence whether a node gets blocked. The engineering team at nthLink remains committed to investigating diverse technologies and undertaking rigorous research to enhance the performance and capabilities of the nthLink system.