Extended Abstract:
I'll Shake Your Hand: What Happens After DNS Poisoning

Jade Sheffey
UMass Amherst
jsheffey@cs.umass.edu

Ali Zohaib
UMass Amherst
azohaib@cs.umass.edu

Dayeon Kang
UMass Amherst
dayeonkang@umass.edu

Zakir Durumeric
Stanford University
zakir@cs.stanford.edu

Amir Houmansadr
UMass Amherst
amir@cs.umass.edu

Qiang Wu
GFW Report
gfw.report@protonmail.com

Abstract

When a DNS request for a censored domain travels across China's network boundary, the Great Firewall (GFW) will inject DNS responses pointing to bogus IP addresses. While packets sent to these IP addresses are often believed to be dropped or null-routed, in this report, we show that for unknown reasons, some of these IP addresses will actually accept TCP handshakes from clients. We characterize this behavior and fingerprint the infrastructure that accepts these client connections. Additionally, we analyze the malformed Teredo addresses sent in response to AAAA queries for censored domains. Finally, we suggest that users encrypt their DNS queries and block all outgoing traffic to these injected IP addresses.

Keywords

internet censorship, Great Firewall of China, DNS

1 Introduction

The Great Firewall of China (GFW) employs DNS packet injection as one of its primary censorship techniques [1, 2, 5, 6, 9]: middleboxes deployed at China's network border monitor network traffic for DNS queries containing blocked domain names. Upon detecting queries to censored domains, middleboxes inject spoofed DNS responses that appear to be from the destination DNS resolver, but contain forged IP addresses [1]. These on-path middleboxes do not block the DNS queries from reaching their destinations, nor do they prevent legitimate responses from reaching the clients. Instead, they rely on network proximity to ensure their forged responses arrive before legitimate ones, as clients typically accept only the first response received for a query [9, §7.2] [2, 9, §7.2].

Prior work has identified at least three different DNS injection systems, each with a unique fingerprint that can be observed through varying IP and DNS protocol characteristics like unique DNS flags (particularly the Authoritative Answer flag), DNS TTL values, IP TTL behavior, and the "Do Not Fragment" IP header flag [2]. Notably, some injectors employ behaviors such as mirroring the TTL value from probe packets in their injected responses, which has implications for common censorship measurement techniques that use TTL-limited packets [3] [2, §4.].

The GFW's DNS injectors send forged IP addresses from specific pools in injected DNS responses, with different groups of addresses used to censor particular sets of domains [2, §3.2] [9, §5.3]. These IP addresses typically belong to organizations outside China, primarily in the United States, including technology companies such as Facebook, Twitter, and Dropbox [2, §3.2] [9, §5.1]. While the domains targeted for censorship evolve over time, the GFW appears to operate primarily through keyword-based filtering, censoring domains containing specific terms rather than maintaining only a static blocklist [1, §6] [9, §5.3]. Previous work [9] discovered that the GFW returns bogus Teredo [11] addresses for IPv6 queries.

In this work, we analyze the characteristics and behavior of the IP addresses returned by the GFW's DNS censorship mechanism. We observe three major findings:

- The malformed Teredo addresses returned by the GFW in response to censored DNS AAAA queries correlate exactly with the IPv4 address pool.
- Eight of the 1922 IPv4 addresses used in injected DNS A responses will complete TCP handshakes when accessed from within China, potentially exposing users to unknown risks.
- Two of these IPv4 addresses host servers that are actively accessible within China, including one hosting a website serving forbidden adult content.

2 Data Collection

To better understand what would happen when a client in China connects to the forged IP addresses provided by the GFW's injectors, we take a two-step approach. In particular, we first acquire a set of forged IP addresses used by the GFW's injectors, then let our client in China connect to each forged IP address.

2.1 DNS Probing

To derive a list of censored domains, we first sent 25 DNS Type A and Type AAAA requests for each domain in the Tranco [12] top 1,000,000 list[1] from a vantage point on Tencent Cloud to a U.S. university server under our control. Because our server was not a DNS resolver, any DNS responses received must be injected by middleboxes. We then captured injected DNS responses sent by the GFW. In this initial stage, we collected 8,139 domains that triggered a censorship response on DNS A queries. AAAA queries produced

Free and Open Communications on the Internet 2025(2), 60–63

[1] Tranco list ID 83NKV, generated on April 18, 2025: https://tranco-list.eu/list/83NKV.

an identical list of domains. All censored requests resulted in either one or two responses, indicating that some requests triggered multiple resolvers [2].

Next, we performed 5,000 DNS A and AAAA queries from the same vantage point using only domains that triggered an injected response in the first stage of data collection. From the responses to these queries, we collected a total of 1922 IPv4 and 622 IPv6 addresses from DNS responses. Of these addresses, all match those in datasets collected in previous works studying the GFW's DNS injector [2, 5, 8].

In our AAAA-record queries for blocked domains, we found 622 distinct IPv6 addresses. Thirty belong to Facebook's `2a03:2880::/32` network, all sharing the same 64-bit interface identifier (IID), `face:b00c:0:25de`. The remaining 592 addresses appear in the Teredo range `2001::/32`. Each Teredo address appears to encode an entry from the IPv4 pool directly into the lower 32 bits of the IPv6 address. We also observe the presence of the address `2001::1`, which does not correspond to an injected IPv4 address. Appendix A contains more details about Teredo behavior.

2.2 TCP Probing

For each injected IP addresses from Section 2.1, we first used ZMap [4] from an Alibaba Cloud server to scan all TCP ports and test for a SYN/ACK response. For IPv6 addresses, we used ZMapv6 [7]. Based on this scan, we found 8 IPv4 and no IPv6 addresses that responded to our probes.

For each of the responding IP addresses, we use Nmap (`-sT`) over all ports to verify TCP handshake behavior. Figure 2 and Table 1 visualize the results of this scan. Open ports are shown in green; other ports are closed. Our analysis reveals three distinct categories of behavior:

2.2.1 Category 1: Actively Used IP addresses.

Two of the injected IP addresses appear to host legitimate services. Specifically, 103.230.123.190 hosts OpenSSH 8.2p1 on port 22, and 103.246.246.144 hosts HTTP on port 80, redirecting to a website containing adult content. We verified that the services hosted at these IP addresses are accessible from both within and outside the GFW. We found no correlation between adult-oriented domains and 103.246.246.144 as a response. These IP addresses complete normal TCP handshakes and maintain connections, suggesting they are actual servers rather than the GFW-spoofed responses.

2.2.2 Category 2: Handshake-Accepting IP addresses.

Six IP addresses (8.7.198.46, 39.109.122.128, 46.82.174.69, 59.24.3.174, 93.46.8.90, and 103.97.3.19) exhibit a unique behavior pattern: they accept TCP handshakes across numerous ports when probed from within China, but show no response when probed from outside China. When these IP addresses complete a handshake, they immediately terminate the connection with RST packets as soon as the client sends the first application data. Two IP addresses: 39.109.122.128 and 103.97.3.19 appear to respond to a smaller fraction of ports.

Fingerprinting these responses revealed consistent patterns at both IP and TCP layers. At the IP layer, we observed IPIDs mirroring the value of the triggering packet. At the TCP layer, its TCP flag follows a simple pattern:

- When a client sends a packet with only the SYN flag set, the censoring machine replies with a SYN+ACK packet.
- When a client sends a packet with the PSH flag set (indicating data transmission), it immediately terminates the connection with a RST packet.

Our experiments indicate that the server's implementation of this behavior is likely stateless — it does not terminate established connections until data is sent, and it does not retransmit SYN+ACK packets when clients don't ACK.

2.2.3 Category 3: Silent IP addresses.

The remaining injected IPv4 addresses did not respond to any probe from within China. These appear to be unreachable or blocked addresses, likely intended to disrupt connections.

3 Limitations

DNS probes were performed from a Tencent server in Guangzhou, and TCP scans were performed from an Alicloud server in Guangzhou. It is possible that injected IP addresses or behaviors vary based on location or ISP [13].

4 Conclusion

The responses we have discovered from IP addresses used in the GFW's DNS poisoning pose a variety of risks to users. One risk is that IP addresses sent by the GFW become active, forwarding censored traffic to servers operated by a third party, such as the IP described in Section 2.2.1 which redirects 0.164% of requests for all censored domains to an adult webpage. The other is that these IP addresses silently complete connections, potentially for the purpose of surveilling China's netizens. To avoid these risks, encrypted DNS solutions such as DoT or DoH can prevent poisoning attacks that put users at risk. Our findings raise several questions warranting further investigation. Measurements from multiple vantage points across different Chinese provinces and ISPs are needed to determine whether the handshake-accepting behavior is uniform or exhibits regional variations. Path analysis using TTL-limited probing could help attribute these hosts to specific infrastructure and determine whether these responses originate from GFW infrastructure.

References

[1] Anonymous. 2014. Towards a Comprehensive Picture of the Great Firewall's DNS Censorship. In Free and Open Communications on the Internet. USENIX. https://www.usenix.org/system/files/conference/foci14/foci14-anonymous.pdf

[2] Anonymous, Arian Akhavan Niaki, Nguyen Phong Hoang, Phillipa Gill, and Amir Houmansadr. 2020. Triplet Censors: Demystifying Great Firewall's DNS Censorship Behavior. In Free and Open Communications on the Internet. USENIX. https://www.usenix.org/system/files/foci20-paper-anonymous_0.pdf

[3] Haixin Duan, Nicholas Weaver, Zongxu Zhao, Meng Hu, Jinjin Liang, Jian Jiang, Kang Li, and Vern Paxson. 2012. Hold-On: Protecting Against On-Path DNS Poisoning. In Securing and Trusting Internet Names. National Physical Laboratory. https://www.icir.org/vern/papers/hold-on.satin12.pdf

[4] Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. 2013. ZMap: Fast Internet-wide Scanning and Its Security Applications. In USENIX Security Symposium. USENIX. https://www.usenix.org/conference/usenixsecurity13/technical-sessions/paper/durumeric

[5] Shencha Fan, Jackson Sippe, Sakamoto San, Jade Sheffey, David Fifield, Amir Houmansadr, Elson Wedwards, and Eric Wustrow. 2025. Wallbleed: A Memory Disclosure Vulnerability in the Great Firewall of China. In Network and Distributed System Security. The Internet Society. https://gfw.report/publications/ndss25/data/paper/wallbleed.pdf

[6] Oliver Farnan, Alexander Darer, and Joss Wright. 2016. Poisoning the Well – Exploring the Great Firewall's Poisoned DNS Responses. In Workshop on Privacy in the Electronic Society. ACM. https://dl.acm.org/authorize?N25517

[7] Oliver Gasser, Quirin Scheitle, Sebastian Gebhard, and Georg Carle. 2016. Scanning the IPv6 Internet: Towards a Comprehensive Hitlist. arXiv:1607.05179 [cs.NI] https://arxiv.org/abs/1607.05179

[8] Nguyen Phong Hoang, Jakub Dalek, Masashi Crete-Nishihata, Nicolas Christin, Vinod Yegneswaran, Michalis Polychronakis, and Nick Feamster. 2024. GFWeb: Measuring the Great Firewall's Web Censorship at Scale. In USENIX Security Symposium. USENIX. https://www.usenix.org/system/files/sec24fall-prepub-310-hoang.pdf

[9] Nguyen Phong Hoang, Arian Akhavan Niaki, Jakub Dalek, Jeffrey Knockel, Pellaeon Lin, Bill Marczak, Masashi Crete-Nishihata, Phillipa Gill, and Michalis Polychronakis. 2021. How Great is the Great Firewall? Measuring China's DNS Censorship. In USENIX Security Symposium. USENIX. https://www.usenix.org/system/files/sec21-hoang.pdf

[10] C. Huitema. 2006. RFC 4380: Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs).

[11] Christian Huitema. 2006. Teredo: Tunneling IPv6 over UDP through network address translations (NATs). Technical Report.

[12] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. 2019. Tranco: A research-oriented top sites ranking hardened against manipulation. In Network and Distributed System Security Symposium 2019 (NDSS '19). https://doi.org/10.14722/ndss.2019.23386

[13] Mingshi Wu, Ali Zohaib, Zakir Durumeric, Amir Houmansadr, and Eric Wustrow. 2025. A Wall Behind A Wall: Emerging Regional Censorship in China. In Symposium on Security & Privacy. IEEE. https://gfw.report/publications/sp25/data/paper/paper.pdf

A Teredo

The Teredo addresses returned by the GFW in the format `2001::WWXX:YYZZ` are fundamentally malformed and do not conform to the Teredo tunneling protocol specification [10]. Teredo addresses must follow a 128-bit structure that encodes specific networking information, and the GFW-injected addresses completely omit or malform numerous components of this structure.

A valid Teredo address has the complete structure

`2001:0000:SSSS:SSSS:FFFF:FFFF:CCCC:CCCC`

where each component serves a specific purpose [10]. The first 32 bits are always `2001:0000`, identifying the Teredo prefix. The next 32 bits (`SSSS:SSSS`) encode the IPv4 address of the Teredo server. The following 16 bits (`FFFF`) contain flags and the cone bit indicating NAT behavior. The final 32 bits (`CCCC:CCCC`) represent the obfuscated (bitwise inverted) external IPv4 address and UDP port of the client behind the NAT.

`Prefix`	Server IPv4	Flags	Port	Client IPv4
2001:0000	0000:0000	0000	0000	67f6:f690
2001:0000	0000:0000	0000	0000	0000:0001

Figure 1: Valid Teredo Address Structure (128 bits total)

Figure 1 shows the structure of a Teredo address with two classes of example injected addresses. In this address, the server IPv4 (0.0.0.0) and port (0), and client port (0) are nonsensical, while the client IPv4 is simply a hex-encoded version of an IP from the IPv4 pool, rather than the standard-compliant method of using an obfuscated version of the target IP. It is likely that these invalid addresses are intended to block access entirely.

B TCP Probe Results

Table 1: Ports that complete a TCP handshake for each responding category 1 IP address.

IP	Ports	Ephemeral Ports
103.246.246.144	80, 3398, 5985, 47001, 49152, 49153, 49154, 49158, 49166	49152, 49153, 49154, 49158, 49166
103.230.123.190	22, 40800, 40810, 40820	n/a

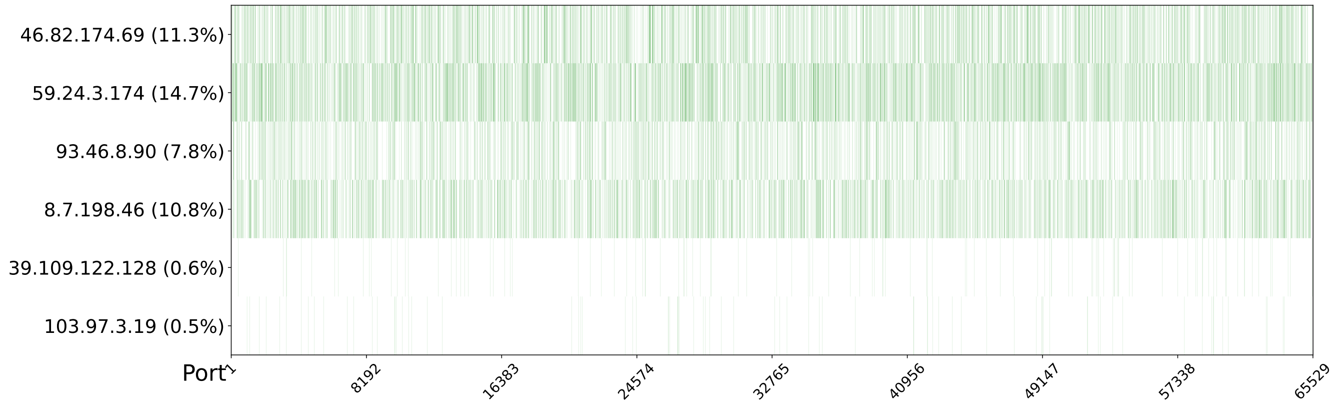

Figure 2: Ports that complete a TCP handshake for each responding category 2 IP address. Percentages indicate the density of responding ports out of 65535