Technical Analysis of the IRGFW

Understanding The Iranian Great Firewall

Report 1

December 2024

IRGFW
irgfw.report

Glossary

Word	Meaning	Word	Meaning	Word	Meaning	Word	Meaning
ASIC	Application-Specific Integrated Circuit	**ECH**	Encrypted Client Hello	**ISP**	Internet Service Provider	**TCP**	Transmission Control Protocol
ASN	Autonomous System Number	**ESNI**	Encrypted Server Name Indication	**L2TP**	Layer 2 Tunneling Protocol	**TIC**	Telecommunication Infrastructure Company
CDN	Content Delivery Network	**GFW**	Great Firewall	**MCI**	Mobile Communication Company of Iran	**TLS**	Transport Layer Security
CIDR	Classless Inter-Domain Routing	**GRPC**	gRPC Remote Procedure Call	**MTN**	Irancell	**TOR**	The Onion Router
CPU	Central Processing Unit	**HTTP**	Hypertext Transfer Protocol	**NIN**	National Information Network	**UDP**	User Datagram Protocol
DDOS	Distributed Denial-of-Service	**HTTPS**	Hypertext Transfer Protocol Secure	**OBFS**	Obfuscation	**UL**	Upload
DL	Download	**HU**	HttpUpgrade *(=WebSocket)*	**P2P**	Peer-to-Peer	**UTLS**	Universal Transport Layer Security
DNS	Domain Name System	**ICMP**	Internet Control Message Protocol	**PPTP**	Point-to-Point Tunneling Protocol	**VPN**	Virtual Private Network
DOH	DNS-over-HTTPS	**IKEV2**	Internet Key Exchange version 2	**QUIC**	Quick UDP Internet Connections	**VPS**	Virtual Private Server
DOQ	DNS-over-QUIC	**IPM**	Institute for Research in Fundamental Sciences	**SNI**	Server Name Indication	**WS**	WebSocket
DOT	DNS-over-TLS	**IPSEC**	Internet Protocol Security	**SSH**	Secure Shell		
DOU	DNS-over-UDP	**IPV4**	Internet Protocol Version 4	**SSTP**	Secure Socket Tunneling Protocol		
DPI	Deep Packet Inspection	**IPV6**	Internet Protocol Version 6	**SYN**	Synchronize		
DTLS	Datagram Transport Layer Security	**IRGFW**	Iranian Great Firewall	**TCI**	Telecommunication Company of Iran		

(Table 1 – Glossary)

Brief History of the Iranian Great Firewall (IRGFW)

Before the tragic death of Mahsa Amini[1][2], the Islamic Regime of Iran's internet filtering system was relatively unsophisticated, the primary methods used were DNS and SNI blocking, which blocked non-TLS and TLS connections to foreign IP addresses. Deep packet inspection *(DPI)* and active probing technologies were minimal and largely invisible, indicating a less comprehensive approach to controlling internet traffic.

The situation drastically changed following Mahsa Amini's death and the subsequent nationwide protests. The Telecommunication Infrastructure Company *(TIC)* and other entities significantly upgraded the nation's internet censorship infrastructure. This involved acquiring and importing advanced firewall and DPI hardware, marking a shift towards a more rigorous and sophisticated internet control system.[3]

Iran has been increasingly inspired by the Chinese internet censorship model, often called the "Great Firewall of China."[4] Despite official claims that Iran is not directly following China's example, there are undeniable parallels in the methods and strategies employed.[5] Iran has developed its national internet infrastructure, aiming to increase domestic Internet traffic to 70% of the total internet traffic in the country, similar to China's promotion of local internet services to reduce reliance on global platforms.[6][7][8]

In addition to hardware upgrades, Iran has imposed stricter regulations on internet platforms, requiring them to comply with local laws or face censorship. This project, called "Sianat" aims to create a controlled internet environment that minimizes the influence of foreign platforms and increases the government's control over digital content and communication.[9]

IRGFW: Digital Boundaries

The IRGFW also features extensive use of DPI to inspect and filter internet traffic at a granular level. This technology allows the government to block specific websites, monitor internet usage, and prevent access to certain content. The primary consumer ISPs in Iran, such as the **Mobile Communication Company of Iran** *(MCI)*, **IranCell** *(MTN)*, and the **Telecommunication Company of Iran** *(TCI)*, connect upstream to the **TIC** *(AS49666)*, which houses the primary firewall. This centralization ensures that blocking and filtering measures are uniformly enforced across all ISPs. [10][11]

The nationwide implementation of these advanced technologies and strict regulatory policies has significantly enhanced Iran's ability to monitor and control internet usage. This transformation reflects a broader trend towards increasing digital authoritarianism, leveraging state-of-the-art technologies to control information and suppress dissent.

The Iranian Great Firewall (IRGFW) is a complex and repressive internet censorship and surveillance apparatus. By employing advanced network filtering and traffic inspection techniques, it enforces pervasive restrictions on online communication and information access. Despite its oppressive nature, it is considered among the world's more formidable national censorship systems, integrating multiple technological and policy layers to strictly regulate and monitor internet traffic within Iran. This report provides a detailed technical analysis of the IRGFW's infrastructure and operational mechanisms to better understand how it achieves its high level of control.

At its core, the IRGFW operates through a coordinated effort involving major Internet Service Providers (ISPs) and the Telecommunication Infrastructure Company *(TIC)*, which serves as the primary upstream provider. Through deep packet inspection *(DPI)*, IP blocking, and other advanced network management techniques, the IRGFW enforces stringent controls over data flow. Understanding the architecture and operation of this system is crucial for comprehending the extent and efficiency of internet censorship in Iran.

First, we need some basic understanding of how and where the IranGFW *(IRGFW)* works. And for sure, we can say it is a unique set of firewalls.

Iran AS Cones and Firewalls

Major consumer Iranian ISPs are:

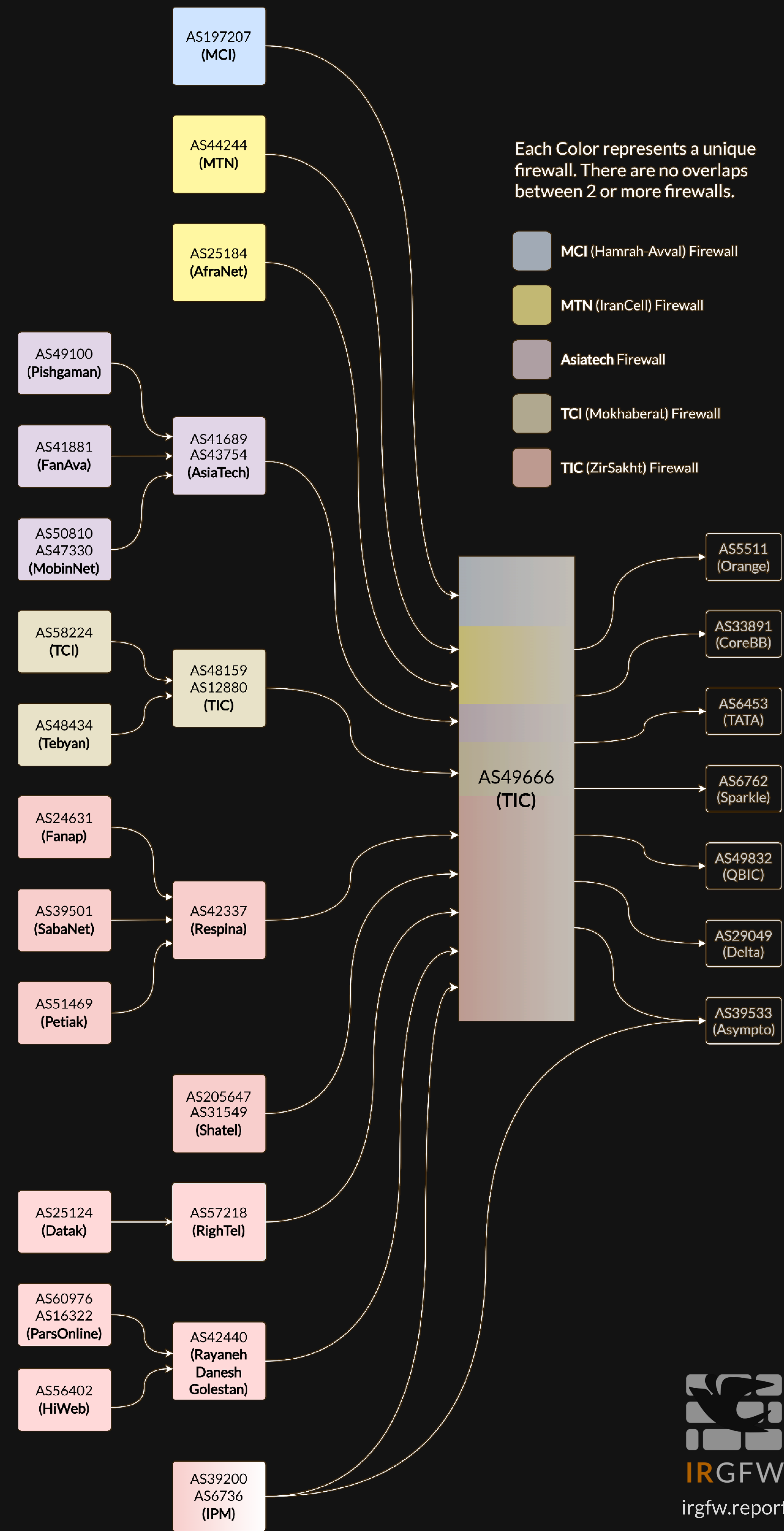
1. **MCI** *(AS197207)* [12]
2. **MTN** *(AS44244)* [13]
3. **TCI** *(AS48159 + AS12880)* [14][15]

All internet operators will go upstream to the **Telecommunication Infrastructure Company** *(TIC)* – **AS49666**, which houses the primary firewall and gateway.[15]

Each of these ISPs has different types of firewalls, and all other operators will follow one of these firewalls. For instance, when an IP address gets blocked on ASIATECH, it is blocked on PISHGAMAN, FANAVA, and MOBINNET. This is true for the DPI and firewall itself as well. The most advanced firewall belongs to the MCI operator *(the biggest mobile operator in Iran)*.

However, these firewalls *(especially MCI)* are sometimes turned off due to countrywide events. The TIC *(AS49666)* primary firewall will be used when the ISP firewall is turned off. For instance, when the MCI firewall is turned on, the TIC firewall will be overridden; thus, when an IP address gets blocked on MCI, it won't be blocked on TIC to some extent. After some time *(based on a "time-pattern" that we discuss in this report)*, the blacklist database will be synced to TIC *(AS49666)*, and then it's blocked on all ISPs.

The IPM *(AS6736)* Internet provides access to free Internet without the restrictions of the primary *(AS49666)* firewall. This organization is one of the oldest and was originally established for elite individuals, government officials, and verified researchers. Additionally, it has a strict bandwidth limit, typically capping at 10/100 Mbps.[16][17]



(Diagram 1 – IR AS Firewalls)
[Link]

IRGFW
irgfw.report

DNS Situation

DNS Situation

DNS requests are subject to DPI, which often results in frequent poisoning and disruptions of DNS queries. Requests to well-known DNS providers are consistently graylisted, regardless of the encryption method used—whether DNS over UDP *(DoU)*, DNS over TLS *(DoT)*, DNS over HTTPS *(DoH)*, or DNS over QUIC *(DoQ)*. This issue is so widespread that, in many cases, it is necessary to rely on the DNS servers provided by the ISP for domestic (local) connections, particularly for traffic routed through the IRGFW.[25]

However, users can mitigate these disruptions by setting up their own DNS servers using encrypted protocols *(DoT, DoH, or DoQ)*. This approach enables users to bypass DNS poisoning, but it introduces two key challenges:

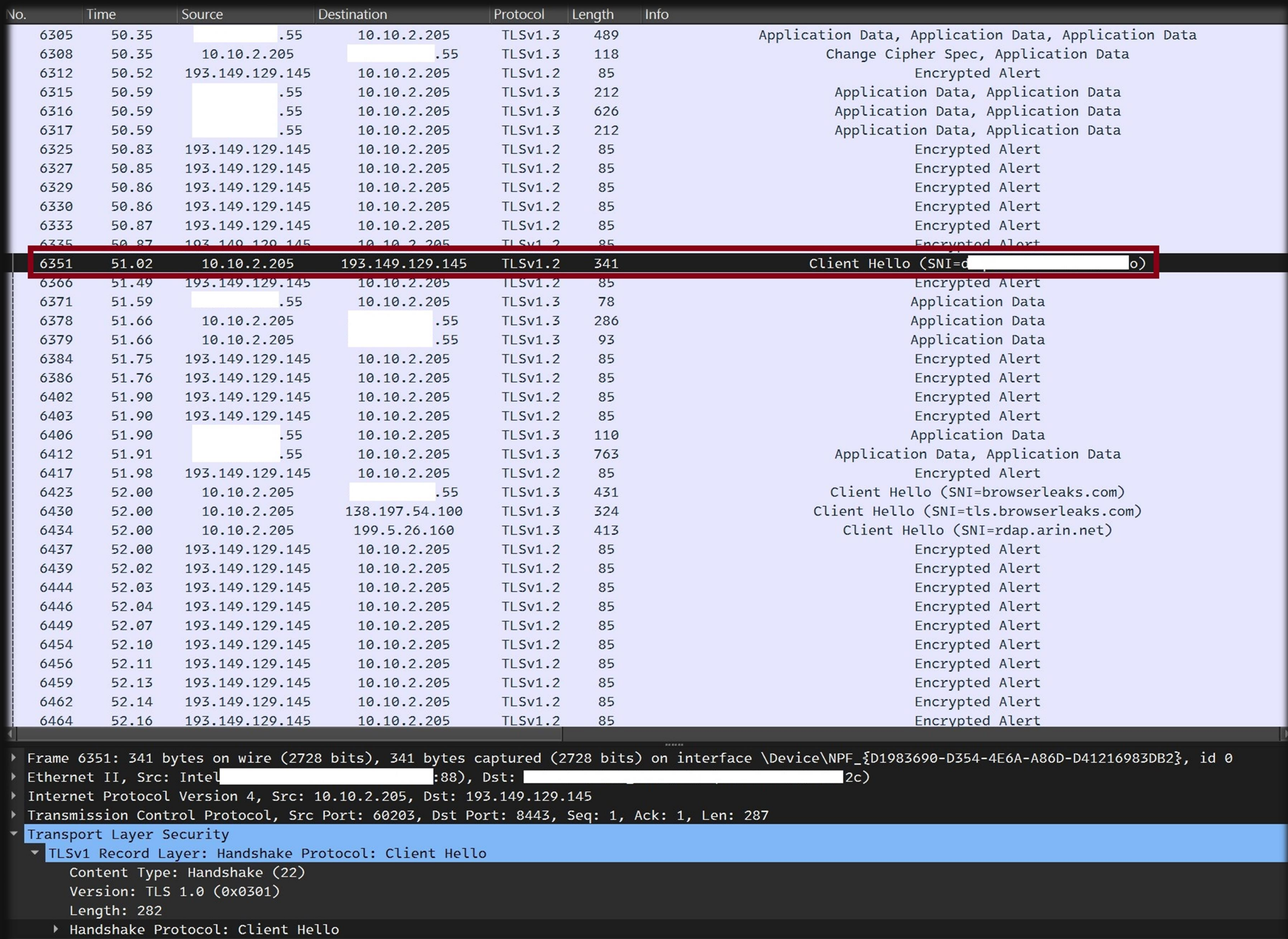
- **Graylisting of Destination IPs:** If the destination IP address is graylisted, the TLS handshake process may fail, preventing the establishment of DoT and DoH connections.
- **Using DoQ:** If UDP traffic is allowed to the destination IP, DoQ can be used to overcome the restrictions associated with the TLS handshake, ensuring secure DNS resolution despite DPI interference.

These challenges underscore the need for advanced DNS management techniques that address both the use of encrypted DNS protocols and the complexities introduced by DPI and graylisting.

DNS Situation

DNS-over-HTTPS *(DoH)*

In this scenario, we configure a DoH server with a whitelisted IP address and an SNI domain. The server listens for DoH requests on both port 443 and port 8443, with Nginx acting as the web server for general HTTP/HTTPS traffic. Both ports are accessible from a standard web browser, allowing the associated website to load normally. However, when using a popular DNS client *(e.g., YogaDNS)*, DoH requests are blocked. Specifically, ClientHello messages are successfully transmitted to the server, but no corresponding ServerHello messages are returned, causing the DoH connection to time out. This behaviour suggests a filtering mechanism affecting the DoH handshake process, preventing the successful resolution of DNS queries over HTTPS.



No.	Time	Source	Destination	Protocol	Length	Info
6305	50.35	[redacted].55	10.10.2.205	TLSv1.3	489	Application Data, Application Data, Application Data
6308	50.35	10.10.2.205	[redacted].55	TLSv1.3	118	Change Cipher Spec, Application Data
6312	50.52	193.149.129.145	10.10.2.205	TLSv1.2	85	Encrypted Alert
6315	50.59	[redacted].55	10.10.2.205	TLSv1.3	212	Application Data, Application Data
6316	50.59	[redacted].55	10.10.2.205	TLSv1.3	626	Application Data, Application Data
6317	50.59	[redacted].55	10.10.2.205	TLSv1.3	212	Application Data, Application Data
6325	50.83	193.149.129.145	10.10.2.205	TLSv1.2	85	Encrypted Alert
6327	50.85	193.149.129.145	10.10.2.205	TLSv1.2	85	Encrypted Alert
6329	50.86	193.149.129.145	10.10.2.205	TLSv1.2	85	Encrypted Alert
6330	50.86	193.149.129.145	10.10.2.205	TLSv1.2	85	Encrypted Alert
6333	50.87	193.149.129.145	10.10.2.205	TLSv1.2	85	Encrypted Alert
6335	50.87	193.149.129.145	10.10.2.205	TLSv1.2	85	Encrypted Alert
6351	51.02	10.10.2.205	193.149.129.145	TLSv1.2	341	Client Hello (SNI=d[redacted]o)
6366	51.49	193.149.129.145	10.10.2.205	TLSv1.2	85	Encrypted Alert
6371	51.59	[redacted].55	10.10.2.205	TLSv1.3	78	Application Data
6378	51.66	10.10.2.205	[redacted].55	TLSv1.3	286	Application Data
6379	51.66	10.10.2.205	[redacted].55	TLSv1.3	93	Application Data
6384	51.75	193.149.129.145	10.10.2.205	TLSv1.2	85	Encrypted Alert
6386	51.76	193.149.129.145	10.10.2.205	TLSv1.2	85	Encrypted Alert
6402	51.90	193.149.129.145	10.10.2.205	TLSv1.2	85	Encrypted Alert
6403	51.90	193.149.129.145	10.10.2.205	TLSv1.2	85	Encrypted Alert
6406	51.90	[redacted].55	10.10.2.205	TLSv1.3	110	Application Data
6412	51.91	[redacted].55	10.10.2.205	TLSv1.3	763	Application Data, Application Data
6417	51.98	193.149.129.145	10.10.2.205	TLSv1.2	85	Encrypted Alert
6423	52.00	10.10.2.205	[redacted].55	TLSv1.3	431	Client Hello (SNI=browserleaks.com)
6430	52.00	10.10.2.205	138.197.54.100	TLSv1.3	324	Client Hello (SNI=tls.browserleaks.com)
6434	52.00	10.10.2.205	199.5.26.160	TLSv1.3	413	Client Hello (SNI=rdap.arin.net)
6437	52.00	193.149.129.145	10.10.2.205	TLSv1.2	85	Encrypted Alert
6439	52.02	193.149.129.145	10.10.2.205	TLSv1.2	85	Encrypted Alert
6444	52.03	193.149.129.145	10.10.2.205	TLSv1.2	85	Encrypted Alert
6446	52.04	193.149.129.145	10.10.2.205	TLSv1.2	85	Encrypted Alert
6449	52.07	193.149.129.145	10.10.2.205	TLSv1.2	85	Encrypted Alert
6454	52.10	193.149.129.145	10.10.2.205	TLSv1.2	85	Encrypted Alert
6456	52.11	193.149.129.145	10.10.2.205	TLSv1.2	85	Encrypted Alert
6459	52.13	193.149.129.145	10.10.2.205	TLSv1.2	85	Encrypted Alert
6462	52.14	193.149.129.145	10.10.2.205	TLSv1.2	85	Encrypted Alert
6464	52.16	193.149.129.145	10.10.2.205	TLSv1.2	85	Encrypted Alert

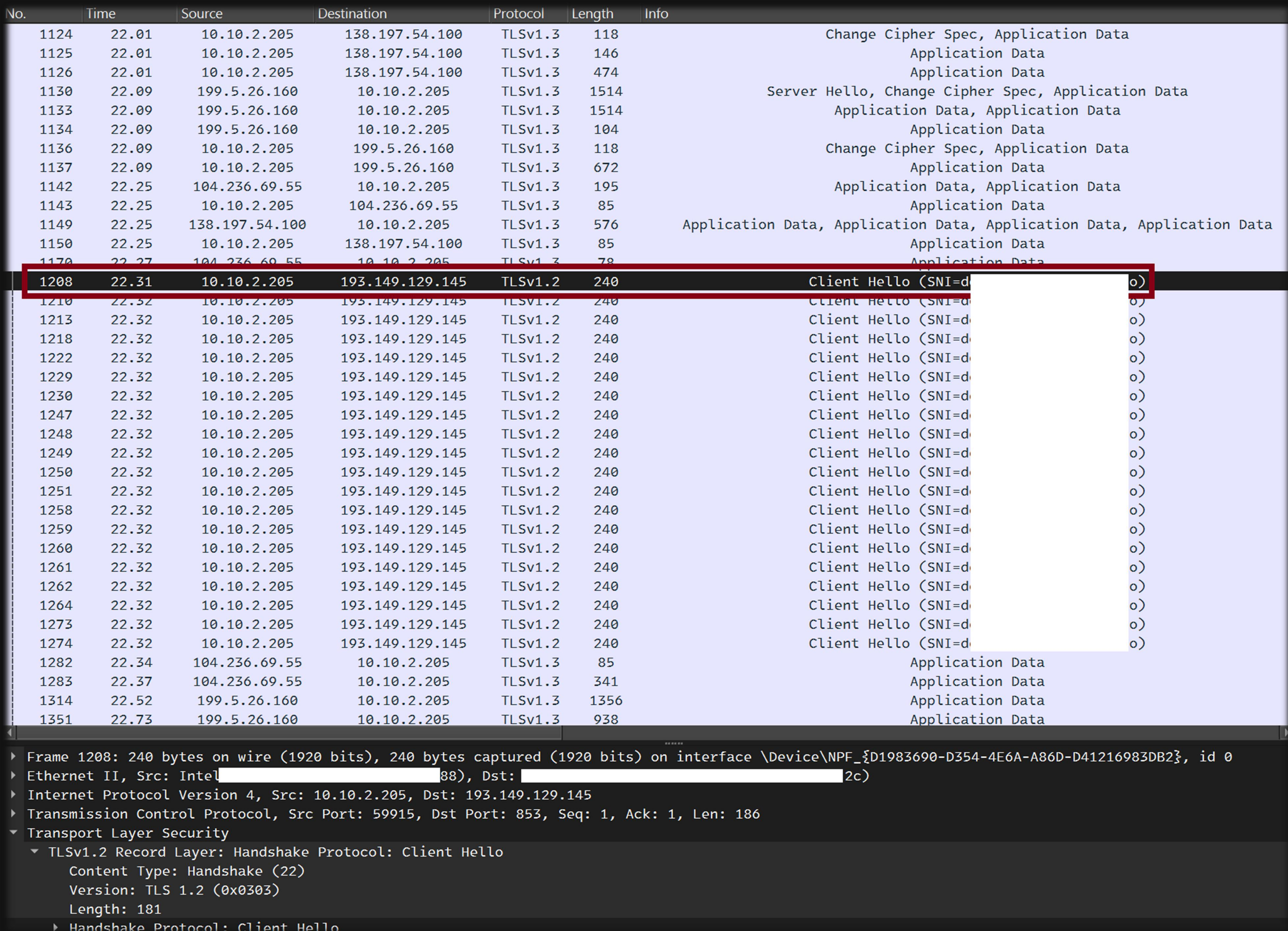
```
Frame 6351: 341 bytes on wire (2728 bits), 341 bytes captured (2728 bits) on interface \Device\NPF_{D1983690-D354-4E6A-A86D-D41216983DB2}, id 0
Ethernet II, Src: Intel[redacted]:88), Dst: [redacted]2c)
Internet Protocol Version 4, Src: 10.10.2.205, Dst: 193.149.129.145
Transmission Control Protocol, Src Port: 60203, Dst Port: 8443, Seq: 1, Ack: 1, Len: 287
Transport Layer Security
  TLSv1 Record Layer: Handshake Protocol: Client Hello
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 282
    Handshake Protocol: Client Hello
```

(Image 1 – DoH)
[Link]

DNS Situation

DNS-over-TLS *(DoT)*

In this scenario, we configure a DoT server to listen on port 853, (which also supports DoQ). A DoT request is sent to the server; however, similar to the previous DoH scenario, the TLS handshake fails. While ClientHello messages are transmitted successfully to the server, no corresponding ServerHello responses are received, resulting in a timeout. This indicates that the TLS negotiation is being blocked or interrupted, preventing the establishment of a secure connection for DNS resolution over TLS.



No.	Time	Source	Destination	Protocol	Length	Info
1124	22.01	10.10.2.205	138.197.54.100	TLSv1.3	118	Change Cipher Spec, Application Data
1125	22.01	10.10.2.205	138.197.54.100	TLSv1.3	146	Application Data
1126	22.01	10.10.2.205	138.197.54.100	TLSv1.3	474	Application Data
1130	22.09	199.5.26.160	10.10.2.205	TLSv1.3	1514	Server Hello, Change Cipher Spec, Application Data
1133	22.09	199.5.26.160	10.10.2.205	TLSv1.3	1514	Application Data, Application Data
1134	22.09	199.5.26.160	10.10.2.205	TLSv1.3	104	Application Data
1136	22.09	10.10.2.205	199.5.26.160	TLSv1.3	118	Change Cipher Spec, Application Data
1137	22.09	10.10.2.205	199.5.26.160	TLSv1.3	672	Application Data
1142	22.25	104.236.69.55	10.10.2.205	TLSv1.3	195	Application Data, Application Data
1143	22.25	10.10.2.205	104.236.69.55	TLSv1.3	85	Application Data
1149	22.25	138.197.54.100	10.10.2.205	TLSv1.3	576	Application Data, Application Data, Application Data, Application Data
1150	22.25	10.10.2.205	138.197.54.100	TLSv1.3	85	Application Data
1170	22.27	104.236.69.55	10.10.2.205	TLSv1.3	78	Application Data
1208	22.31	10.10.2.205	193.149.129.145	TLSv1.2	240	Client Hello (SNI=d[redacted]o)
1210	22.32	10.10.2.205	193.149.129.145	TLSv1.2	240	Client Hello (SNI=d[redacted]o)
1213	22.32	10.10.2.205	193.149.129.145	TLSv1.2	240	Client Hello (SNI=d[redacted]o)
1218	22.32	10.10.2.205	193.149.129.145	TLSv1.2	240	Client Hello (SNI=d[redacted]o)
1222	22.32	10.10.2.205	193.149.129.145	TLSv1.2	240	Client Hello (SNI=d[redacted]o)
1229	22.32	10.10.2.205	193.149.129.145	TLSv1.2	240	Client Hello (SNI=d[redacted]o)
1230	22.32	10.10.2.205	193.149.129.145	TLSv1.2	240	Client Hello (SNI=d[redacted]o)
1247	22.32	10.10.2.205	193.149.129.145	TLSv1.2	240	Client Hello (SNI=d[redacted]o)
1248	22.32	10.10.2.205	193.149.129.145	TLSv1.2	240	Client Hello (SNI=d[redacted]o)
1249	22.32	10.10.2.205	193.149.129.145	TLSv1.2	240	Client Hello (SNI=d[redacted]o)
1250	22.32	10.10.2.205	193.149.129.145	TLSv1.2	240	Client Hello (SNI=d[redacted]o)
1251	22.32	10.10.2.205	193.149.129.145	TLSv1.2	240	Client Hello (SNI=d[redacted]o)
1258	22.32	10.10.2.205	193.149.129.145	TLSv1.2	240	Client Hello (SNI=d[redacted]o)
1259	22.32	10.10.2.205	193.149.129.145	TLSv1.2	240	Client Hello (SNI=d[redacted]o)
1260	22.32	10.10.2.205	193.149.129.145	TLSv1.2	240	Client Hello (SNI=d[redacted]o)
1261	22.32	10.10.2.205	193.149.129.145	TLSv1.2	240	Client Hello (SNI=d[redacted]o)
1262	22.32	10.10.2.205	193.149.129.145	TLSv1.2	240	Client Hello (SNI=d[redacted]o)
1264	22.32	10.10.2.205	193.149.129.145	TLSv1.2	240	Client Hello (SNI=d[redacted]o)
1273	22.32	10.10.2.205	193.149.129.145	TLSv1.2	240	Client Hello (SNI=d[redacted]o)
1274	22.32	10.10.2.205	193.149.129.145	TLSv1.2	240	Client Hello (SNI=d[redacted]o)
1282	22.34	104.236.69.55	10.10.2.205	TLSv1.3	85	Application Data
1283	22.37	104.236.69.55	10.10.2.205	TLSv1.3	341	Application Data
1314	22.52	199.5.26.160	10.10.2.205	TLSv1.3	1356	Application Data
1351	22.73	199.5.26.160	10.10.2.205	TLSv1.3	938	Application Data

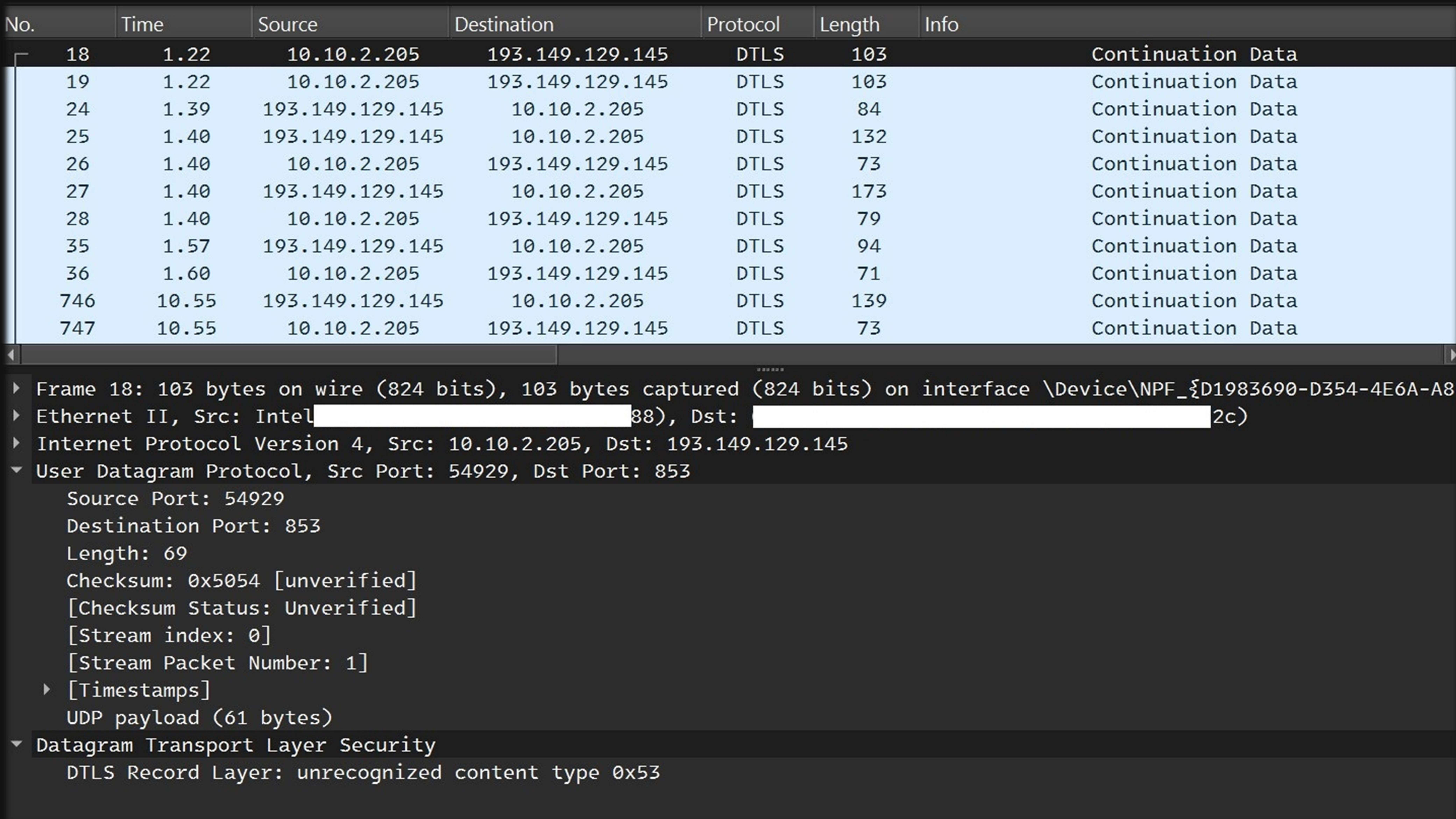
```
Frame 1208: 240 bytes on wire (1920 bits), 240 bytes captured (1920 bits) on interface \Device\NPF_{D1983690-D354-4E6A-A86D-D41216983DB2}, id 0
Ethernet II, Src: Intel[redacted]88), Dst: [redacted]2c)
Internet Protocol Version 4, Src: 10.10.2.205, Dst: 193.149.129.145
Transmission Control Protocol, Src Port: 59915, Dst Port: 853, Seq: 1, Ack: 1, Len: 186
Transport Layer Security
  TLSv1.2 Record Layer: Handshake Protocol: Client Hello
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 181
    Handshake Protocol: Client Hello
```

(Image 2 – DoT)
[Link]

DNS Situation

DNS-over-QUIC *(DoQ)*

In this scenario, DoQ requests are transmitted using the Datagram Transport Layer Security *(DTLS)* protocol, which operates over UDP but provides encryption similar to TLS. Since the DoQ requests are encapsulated in DTLS, they bypass the traditional filtering mechanisms of the IRGFW— as the firewall does not yet recognize this DTLS fingerprint from this specific client. Consequently, the connection is established successfully without disruption, allowing DNS queries to be resolved over QUIC without interference.



No.	Time	Source	Destination	Protocol	Length	Info
18	1.22	10.10.2.205	193.149.129.145	DTLS	103	Continuation Data
19	1.22	10.10.2.205	193.149.129.145	DTLS	103	Continuation Data
24	1.39	193.149.129.145	10.10.2.205	DTLS	84	Continuation Data
25	1.40	193.149.129.145	10.10.2.205	DTLS	132	Continuation Data
26	1.40	10.10.2.205	193.149.129.145	DTLS	73	Continuation Data
27	1.40	193.149.129.145	10.10.2.205	DTLS	173	Continuation Data
28	1.40	10.10.2.205	193.149.129.145	DTLS	79	Continuation Data
35	1.57	193.149.129.145	10.10.2.205	DTLS	94	Continuation Data
36	1.60	10.10.2.205	193.149.129.145	DTLS	71	Continuation Data
746	10.55	193.149.129.145	10.10.2.205	DTLS	139	Continuation Data
747	10.55	10.10.2.205	193.149.129.145	DTLS	73	Continuation Data

```
Frame 18: 103 bytes on wire (824 bits), 103 bytes captured (824 bits) on interface \Device\NPF_{D1983690-D354-4E6A-A8
Ethernet II, Src: Intel[redacted]88), Dst: [redacted]2c)
Internet Protocol Version 4, Src: 10.10.2.205, Dst: 193.149.129.145
User Datagram Protocol, Src Port: 54929, Dst Port: 853
    Source Port: 54929
    Destination Port: 853
    Length: 69
    Checksum: 0x5054 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
    [Stream Packet Number: 1]
    [Timestamps]
    UDP payload (61 bytes)
Datagram Transport Layer Security
    DTLS Record Layer: unrecognized content type 0x53
```

(Image 3 – DoQ)
[Link]

UDP Situation

UDP Situation

First, we should separate the regular UDP and the Unidentified UDP *(or Unknown UDP)*. Regular UDP or UDP generally has fingerprints and identification, like Skype, Zoom and Facetime video calls. Also, a normal Wireguard handshake is based on known UDP; thus, it's easily identifiable and fingerprinted.[18][19]

On the other hand, Unknown UDP refers to UDP handshake or traffic that cannot be immediately recognized or matched to a known application or traffic by network monitoring and security tools. This type of traffic is often characterized by its need for more identifiable signatures, making it challenging to determine its purpose or source.

Unknown-UDP Characteristics:

- **Obfuscation:** Used to hide true traffic nature, common in VPN services like obfuscated WireGuard.
- **Proprietary Protocols:** Custom applications using unique communication methods.
- **Encryption:** Encrypted traffic does not match known patterns, often seen in P2P applications and security tools.

UDP Situation

In I.R. Iran, WireGuard handshakes to foreign IPs are likely blocked through the IRGFW by silently dropping UDP packets when it detects what appears to be a standard WireGuard handshake.[20] In some cases, rate limiting may also be applied to degrade the performance of such connections. This blocking mechanism can potentially be bypassed by adding noise or simulating other handshakes *(or any other known bytes)* before the actual WireGuard handshake. The firewall seems to rely on identifying specific byte patterns in the handshake rather than employing complex regex or deep inspection techniques, which may allow for obfuscation to evade detection.

The firewall appears to buffer or DPI up to 17 KB of UDP *(and TCP as well)* traffic connection per IP:Port combination, analyzing this data to detect WireGuard-specific fingerprints. Beyond this buffer, further traffic is not inspected. The blocking mechanism seems to target high UDP ports (above 1024), while widely used ports like 443 generally remain unaffected. This focus on high ports might make typical WireGuard configurations more susceptible to inspection.

Although UDP is stateless, firewalls often maintain a temporary "connection-like" state for UDP traffic. For example, they associate packets with an IP:Port pair and treat it as a pseudo-session for a limited time *(typically five seconds)*. This state allows the firewall to monitor multiple packets in a flow and identify patterns, such as a WireGuard handshake.

One possible approach to mitigate detection could involve implementing variable-interval port hopping[21], where the port changes at randomized time intervals. This might reduce the likelihood of fingerprinting by introducing unpredictability into traffic patterns. Additionally, altering handshake patterns dynamically and obfuscating payloads may further complicate the firewall's ability to effectively identify and block WireGuard traffic.

Over time, the firewall appears to adapt by recognizing and blocking specific handshake patterns, particularly in cases where repeated traffic is observed between specific IP ranges or data centers. This behaviour suggests the possibility that the firewall can learn and respond to repeated patterns. These observations underline the need for continued experimentation with obfuscation techniques and randomized traffic behaviour to maintain reliable connectivity and potentially outpace the firewall's evolving detection capabilities.

UDP Situation

In this scenario, we observe that a standard WireGuard handshake is consistently being blocked by the firewall at the packet level, occurring at intervals of every 5 seconds. Notably, this 5-second interval is not part of a KeepAlive mechanism, as it's explicitly disabled in this configuration.

Despite the firewall's targeted blocking of the WireGuard handshake, there are no ICMP error messages observed, and the destination IP address continues to respond to ping requests without issue. This behaviour indicates that while the firewall is specifically filtering out WireGuard handshake packets, it does not interfere with general network traffic such as ICMP, ensuring basic connectivity checks remain operational.

No.	Time	Source	Destination	Protocol	Length	Info	Dest Port
7657	9.33	62.[redacted]	45.13[redacted]	WireGuard	190	Handshake Initiation, sender=0xAB8758DE	54571
8523	14.33	62.[redacted]	45.13[redacted]	WireGuard	190	Handshake Initiation, sender=0xF55CBEA3	54571
9241	19.33	62.[redacted]	45.13[redacted]	WireGuard	190	Handshake Initiation, sender=0xD2F7CD1F	54571
12590	24.34	62.[redacted]	45.13[redacted]	WireGuard	190	Handshake Initiation, sender=0xB11AB00C	54571
13709	29.34	62.[redacted]	45.13[redacted]	WireGuard	190	Handshake Initiation, sender=0xB40F9646	54571
17883	34.34	62.[redacted]	45.13[redacted]	WireGuard	190	Handshake Initiation, sender=0xE3F156D5	54571
21228	39.34	62.[redacted]	45.13[redacted]	WireGuard	190	Handshake Initiation, sender=0x352C1132	54571
22784	44.34	62.[redacted]	45.13[redacted]	WireGuard	190	Handshake Initiation, sender=0x410EF8D6	54571
26031	49.34	62.[redacted]	45.13[redacted]	WireGuard	190	Handshake Initiation, sender=0xD8D20074	54571
28588	54.34	62.[redacted]	45.13[redacted]	WireGuard	190	Handshake Initiation, sender=0xBF418730	54571
30795	59.35	62.[redacted]	45.13[redacted]	WireGuard	190	Handshake Initiation, sender=0x1A46B998	54571
32225	64.35	62.[redacted]	45.13[redacted]	WireGuard	190	Handshake Initiation, sender=0x35928FC8	54571
35004	69.35	62.[redacted]	45.13[redacted]	WireGuard	190	Handshake Initiation, sender=0x9E962A9A	54571

(Image 4 – Normal Wireguard)
[Link]

UDP Situation

In this case study, we observe the behaviour of a firewall that actively blocks both the WireGuard handshake and ICMP communication to the destination IP.

Initially, we send minimal "junk" and "noise" packets to simulate traffic before attempting the WireGuard handshake. These packets accumulate in the firewall buffer up to packet number 9472 without triggering any response. Following this, we initiate a QUIC handshake, which successfully bypasses the firewall buffer by sending similar pre-handshake noise and junk packets. This behaviour demonstrates the firewall's ability to handle and scrutinize WireGuard traffic differently compared to QUIC.

The critical observation here is that when the firewall blocks the WireGuard handshake, it also prevents ICMP *(ping)* communication to the destination IP address. This simultaneous blocking indicates a stringent firewall policy that disrupts both VPN handshakes and ICMP traffic.

Importantly, the ICMP connectivity check here is independent of the WireGuard protocol's usual KeepAlive mechanism. Instead, a separate program is employed to test the ICMP connection *(although in Wireshark it's written "Port unreachable", but it results in timeouts in normal ping command)*, isolating the behaviour of the firewall towards diagnostic traffic alongside VPN handshakes.

No.	Time	Source	Destination	Protocol	Length	Info	Dest Port
4163	3.33	45.	6	WireGuard	190	Handshake Initiation, sender=0xF787EFB4	58040
4164	3.33	6	45	ICMP	218	Destination unreachable (Port unreachable)	
6152	8.44		6	WireGuard	190	Handshake Initiation, sender=0xD1BB7F12	58040
6153	8.44	6	45	ICMP	218	Destination unreachable (Port unreachable)	
6875	13.68		6	WireGuard	190	Handshake Initiation, sender=0xD373E425	58040
6876	13.68	6	45	ICMP	218	Destination unreachable (Port unreachable)	
7630	18.90		6	WireGuard	190	Handshake Initiation, sender=0x585838EB	58040
7631	18.90	6	45	ICMP	218	Destination unreachable (Port unreachable)	
8653	24.22		6	WireGuard	190	Handshake Initiation, sender=0xF8CBB029	58040
8654	24.22	6	45	ICMP	218	Destination unreachable (Port unreachable)	
9472	27.40	6	45	QUIC	65	Handshake, DCID=c55c844ce8700531[Malformed Packet]	35197
9474	27.41	6	45	QUIC	65	Handshake, DCID=c55c844ce8700531[Malformed Packet]	35197
9476	27.41	6	45	QUIC	67	Protected Payload (KP0)	35197
9478	27.42	6	45	QUIC	67	Protected Payload (KP0)	35197
9479	27.43	6	45	QUIC	69	Protected Payload (KP0)	35197
9481	27.44	6	45	QUIC	70	Protected Payload (KP0)	35197
9482	27.44	6	45	QUIC	65	Handshake, DCID=b6d42c6c7177df70[Malformed Packet]	35197
9484	27.44	6	45	QUIC	68	Protected Payload (KP0)	35197
9485	27.44	6	45	QUIC	65	Handshake, DCID=c55c844ce8700531[Malformed Packet]	35197
9486	27.44	6	45	QUIC	65	Handshake, DCID=c55c844ce8700531[Malformed Packet]	35197
9487	27.44	6	45	QUIC	69	Protected Payload (KP0)	35197

(Image 5 – Modified Wireguard)

[Link]

QUIC Situation

In Iran, the deployment and utilization of QUIC and HTTP/3 protocols face significant challenges due to stringent government filtering policies. Although HTTP/3 has been partially adopted, its performance is severely throttled, leading to slower speeds than HTTP/2. QUIC handshake/traffic to many international data centers is often blocked, impacting performance inconsistently depending on the destination IP range.

Users attempting to circumvent these restrictions with tools that use QUIC as a tunnelling proxy but experience varying success, as the effectiveness of these tools heavily relies on the specific foreign IP addresses being accessed. Consequently, while these proxy tools can sometimes provide faster and more secure connections, their reliability is significantly based on Iran's pervasive and unpredictable filtering practices.[22]

In addition to these limitations, it has been observed that QUIC traffic to certain foreign IP ranges may be blocked selectively within the same data center, where some IPs remain accessible while others are entirely restricted.[23] This filtering appears to target QUIC handshakes, with specific byte patterns being flagged and blocked after repeated use. For example, frequent QUIC handshakes from Iranian IPs to a particular foreign IP can lead to a complete block on that connection. The filtering mechanism also demonstrates an ability to adapt and block high-frequency QUIC traffic originating from specific IPs after reaching a threshold of traffic volume or repeated patterns. Furthermore, QUIC traffic to Cloudflare has recently declined significantly, potentially indicating targeted restrictions against its widespread use. [24]

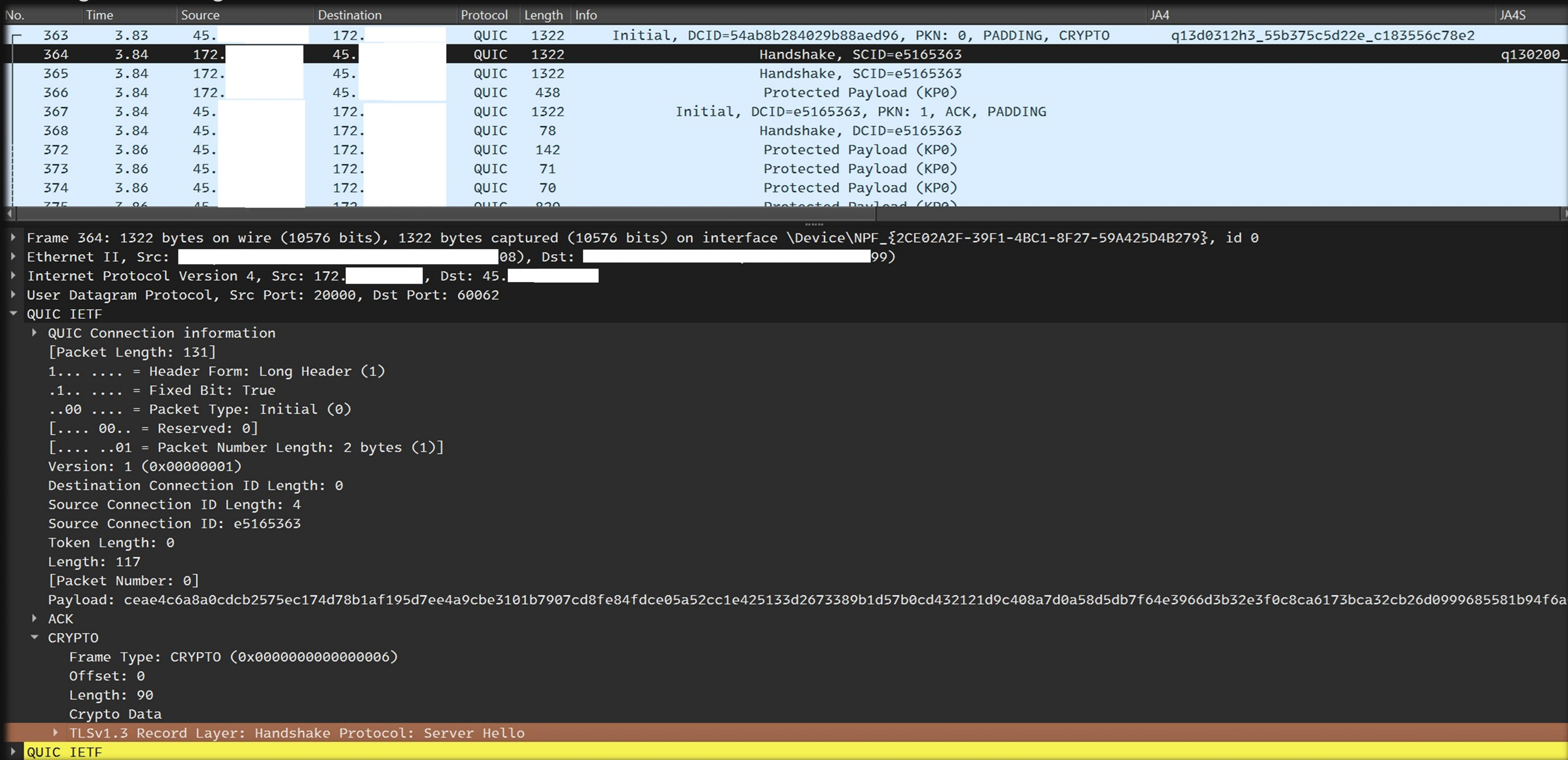
To address these challenges, tools relying on QUIC need to introduce dynamic handshake and traffic obfuscation mechanisms to evade identification by Iranian DPI systems. Adjusting handshake patterns or introducing randomness into QUIC traffic flows may help improve their effectiveness against these restrictions.

QUIC Situation

In this scenario, we tested connectivity to a domain with a specific destination IP where UDP and QUIC traffic are unrestricted. The handshake process was observed in Wireshark, confirming the following sequence:

1. The ClientHello was sent from the client.
2. The ServerHello was received, completing the QUIC handshake.
3. Application-layer payloads were successfully exchanged without any interruptions.

The target server is running a Nginx with HTTP/3 *(QUIC)* support enabled by default. Both curl with HTTP/3 and Hysteria2 were used to validate connectivity and handshake consistency. The results confirm that this domain and IP are fully operational for QUIC traffic, with no evidence of filtering or throttling.



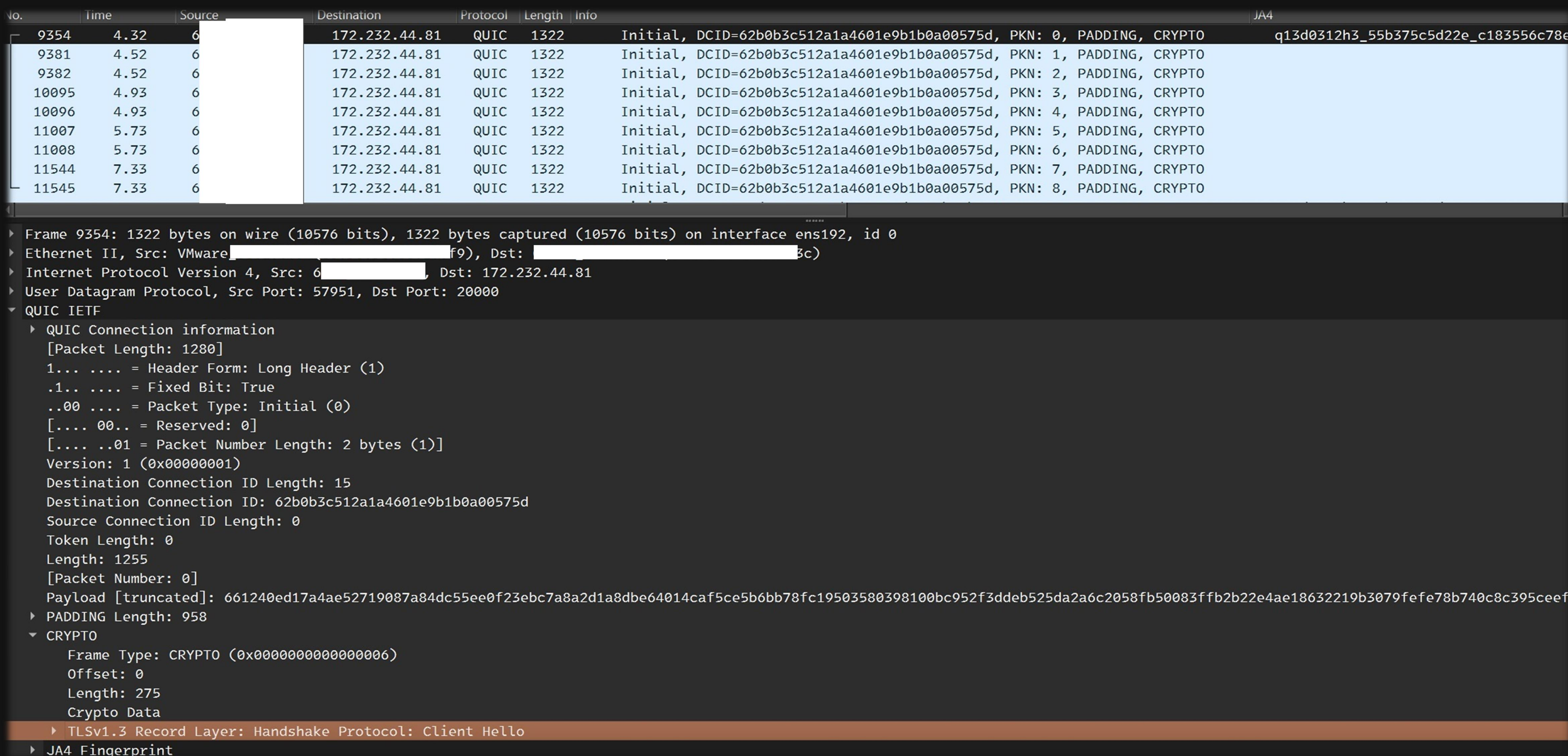
(Image 6 – QUIC Handshake OK)
[Link]

QUIC Situation

In this specific scenario, the destination IP address can be connected with an obfuscated Wireguard, indicating UDP traffic is not blocked to this IP. Then we attempt to initiate a QUIC handshake with a whitelisted domain in Iran. Despite UDP traffic successfully reaching the destination IP, the QUIC handshake fails to complete.

When analyzing the traffic in Wireshark, we observe the client sending the ClientHello. However, all subsequent ClientHello packets are retransmissions, indicating that the client is not receiving a response from the server. No ServerHello is observed or received by the client, which confirms that the handshake is being disrupted after the initial client transmission.

This pattern highlights a filtering mechanism that allows UDP packets through but actively blocks the QUIC handshake process at a protocol-specific level. Such targeted behaviour underscores the sophistication of the filtering system and the need for advanced obfuscation techniques to bypass these restrictions. However, when testing with a non-blocked domain, the blockage consists.



(Image 7 – QUIC Handshake NotOK)
[Link]

IP Address Situation

IP Address Situation

IRGFW has three lists: **WhiteList**, **GrayList**, and **BlackList**. The history of an IP address is a significant factor in this matter.

- **White IP:** The IP should be from a relatively unknown data center; no one has used it for VPN/Proxy for the last three months. *(Or more!)* It should also be manually whitelisted on ISP databases. Thus, sometimes, an extremely unknown data center IP address could be blocked faster because it has not been whitelisted in the IRGFW database.
- **Gray IP:** The IRGFW designates specific IP addresses as "gray" when suspected of being used for VPN or proxy purposes but lacking sufficient evidence to warrant an immediate block. These IP addresses, often belonging to major data centers, are subject to periodic traffic analysis and data collection, likely contributing to limited upload speeds and high jitter. By default, the IRGFW categorizes an IP address as gray and continuously monitors it, gathering traffic samples. Based on the collected data and observed usage patterns over time, the IRGFW will decide whether to block the IP address permanently.
- **Black IP:** After analyzing sufficient data from Gray IPs, the IRGFW may escalate an IP address to Black IP status. This results in complete or partial blockage using different patterns:
    - **TIC and TCI:** These patterns block all types of traffic to the IP, including ICMP, SSH, TLS*(v1.0~v1.3)*, HTTP, and others.
    - **MCI:** When the firewall is active, it explicitly blocks the ServerHello phase of the TLS handshake, disrupting secure connections.
    - **MTN:** This pattern inconsistently blocks traffic, sometimes targeting SSH and TLS protocols and only TLS.

These strategies are part of the IRGFW's comprehensive approach to controlling and limiting VPNs and proxies within the country.

IPv6 Situation

IPv6 has not yet reached mainstream adoption across most operators. However, it is available for mobile users on networks like MCI and MTN, provided the user manually enables it. On these IPv6 addresses, DPI is typically disabled by default, making them less scrutinized. Nevertheless, the fundamental IRGFW rules—such as categorizing IP addresses into WhiteList, GrayList, and BlackList—still apply, though with less stringent enforcement compared to IPv4.

Time Pattern

We have identified specific patterns related to block timings. The TIC primary firewall synchronizes daily at 6:00 AM and 12:00 PM (UTC+03:30). Consequently, when referring to a TIC firewall test, it implies that the TIC will block the servers exclusively during these synchronization periods. In contrast, the MCI firewall may block an IP address or domain at any time during the day, following its time-based patterns.

For clarity, "moderate" traffic is defined as symmetrical traffic of 100 Mbps on the server.

- **Time pattern 1:** 4h - 1d - 4d - 1w - 40d
- **Time pattern 2:** 1h - 4h - 2d - 2w - 40d

1. **Time Pattern 1:** Set up a proxy server with Xray-core like VLESS-TCP-Reality*(Vision) (Combination is unimportant)*. Flow some moderate traffic on it. If the IP address didn't block after 4 hours, it will likely work for 1 day *(The TIC firewall test)*. If the IP has not been blocked, it will likely work for 4 days *(Another TIC firewall test)*. And if it is not blocked yet, it will probably go for 1 week *(The MCI firewall test)*. If passed, it would likely work for 40 days, but after this period, there were so many random factors that we couldn't find any patterns.

2. **Time Pattern 2:** Set up a proxy server like the above. Flow some moderate traffic on it. If the IP address didn't block after 1 hour, it will likely work for 4 hours. If the IP has not been blocked, it will likely work for 2 days *(TIC firewall test)*. And if it is not blocked yet, it will probably go for 2 weeks *(The MCI firewall test)*. If passed, it would likely work for 40 days, but after this period, there were so many random factors that we couldn't find any patterns.

When an IP address is Graylisted, it will never go to Whitelist again! So, when IRGFW throttles the IP address, we can say the IP is gray, and when the IP is blocked, it is in BlackList. Most of the time, after 40 days, the IP will be unblocked again, but now the IP is gray and may have some limitations on DL/UL speed and high jitter in some cases. This pattern will occur for every foreign IP address range, primarily for famous data centers and hosting services that can be used for VPN/Proxy servers; or too infamous ASNs that are not in the default firewalls whitelists.

This "gray-listing" can be used for protocols as well. As we discussed, HTTP3/QUIC and UDP are Graylisted by default unless the client's fingerprint *(e.g. User-Agent in HTTP handshakes or UTLS in Client-Hello)* does not match any of the firewall databases and the destination IP address has not been graylisted yet.

Active-Probes

Active-Probes

IRGFW "had" an active-probing system back in September 2023, and we extracted some of the used IP addresses.[26] Some of our test servers were even impacted by various DDoS attacks, which maxed out the server CPU usage.[27] These IP addresses were handled in these tests on the server using Xray-core.

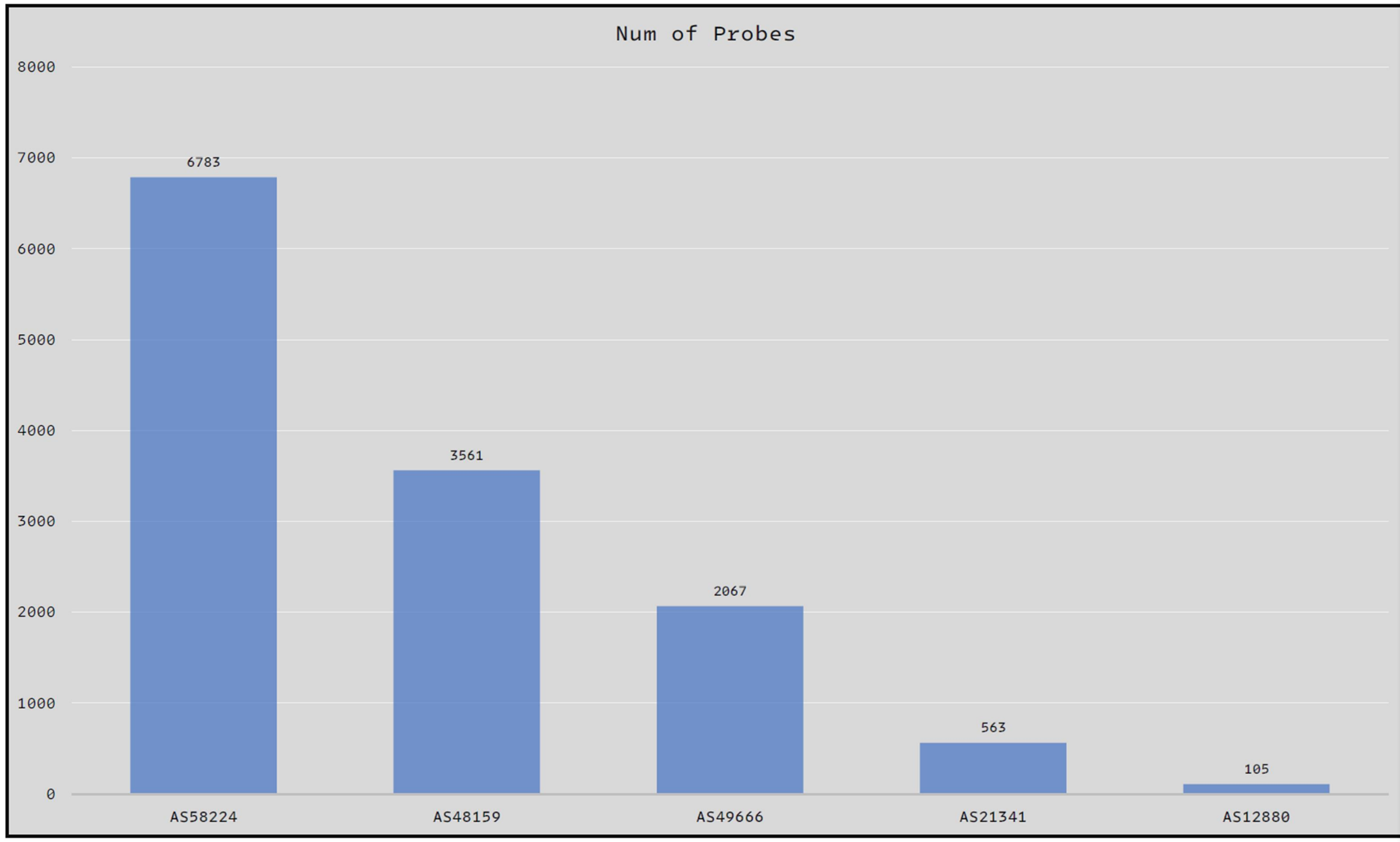
But from early January 2024, IRGFW no longer uses Active-Probes. There are no signs of probes on any servers, and we guess they upgraded the IRGFW to be more precise and optimized on the passive side, as we'll discuss in this report.

In the image below we recorded most of the IP CIDRs that we detected as Probers. Our test method is inspired by gfw.report team.[28]

In the following pages, we cumulated all of our Active-probe tests into **three types**. Most tests were done with Xray-core and others with various cores and methods in Iran.

Iran's GFW Active Probing Test

AS Number	Org/Name	Num of Probes	IP Ranges
AS58224	*Iran Telecommunication Company PJS (TCI)*	6783	80.191.0.0/16 *(80.191.69.0/24)* *(80.191.64.0/24)* 78.38.0.0/16 78.39.0.0/16 217.218.0.0/16 *(217.218.80.0/16)* 2.187.0.0/16
AS48159	*Telecommunication Infrastructure Company (TIC-AS)*	3561	2.189.42.0/24 2.184.0.0/16
AS49666	*Telecommunication Infrastructure Company (TIC-GW-AS)*	2067	2.188.28.0/24
AS21341	*Soroush Rasanheh Company Ltd (SINET-AS)*	563	62.220.121.0/24
AS12880	*Information Technology Company (ITC) (DCI-AS)*	105	2.188.170.0/24



(Image 8 – APO)
[Link]

Active-Probes

Test Type 1

Here, The server, utilizing VLess-TCP-Reality protocol *(Port 2053)*, operated for 24 hours, transferring ~2TB of data before being blocked. Legitimate SYN requests grew steadily, peaking at 332 in hour 19. However, probing activity—*likely from the IRGFW*—began at hour 5, with a sharp increase during hour 16 *(64 probe SYNs alongside 301 legitimate SYNs)*. This suggests deliberate targeting as part of censorship enforcement mechanisms.

Key Observations:

- The Iranian firewall's probes escalated alongside traffic, indicating active surveillance and filtering efforts targeting circumvention tools.
- Despite blocking IR domains and IPs, the server was overwhelmed due to insufficient fallback mechanisms *(e.g., Nginx fallback)* and the absence of adaptive defensive strategies.
- Traffic and probe spikes during hours 16–19 reflect a coordinated probing strategy, likely aiming to detect and disrupt encrypted communication methods.

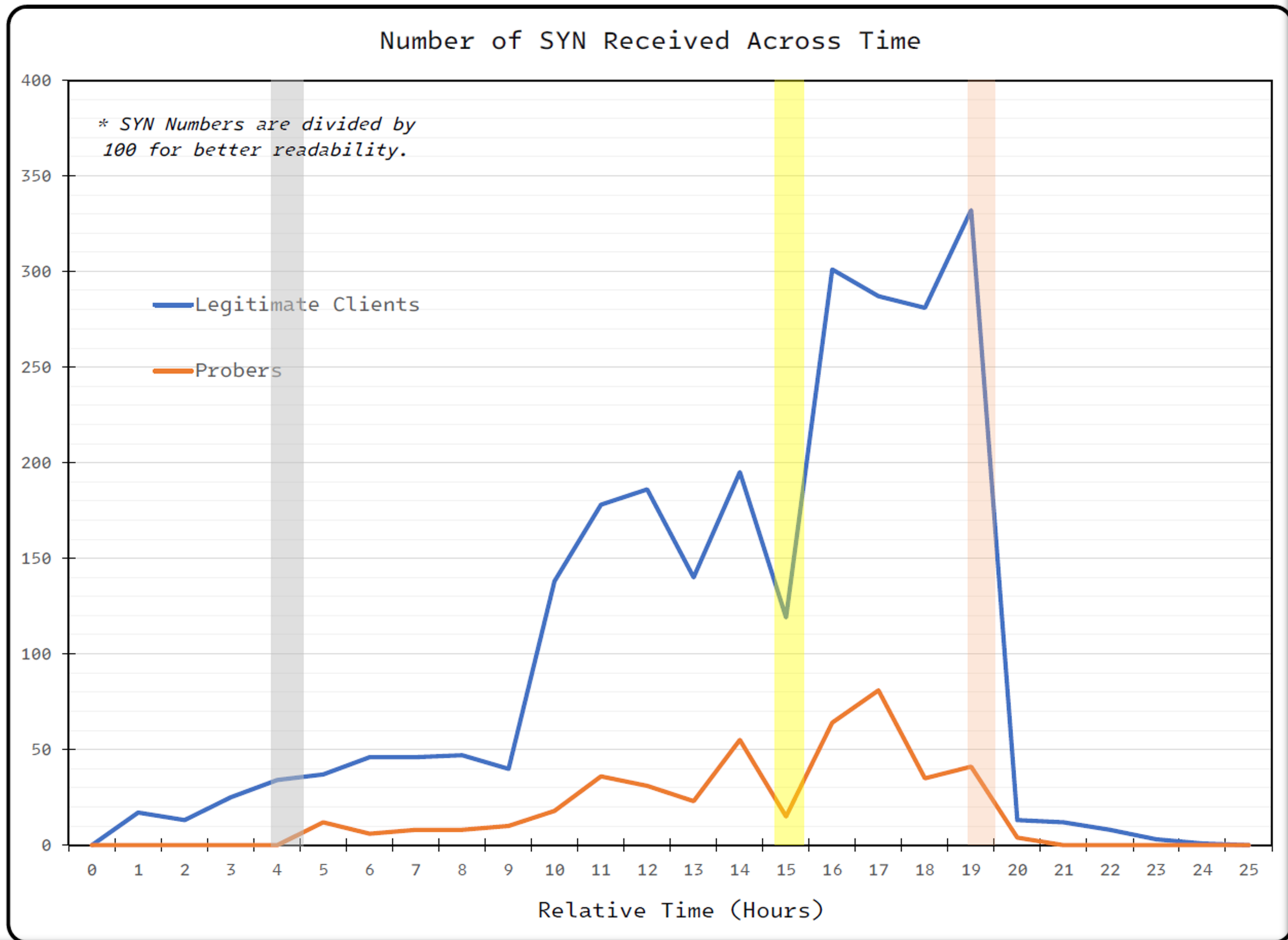
Number of SYN Received Across Time

Relative Time (Hours)	*Legitimate Clients*	*Probers*	
0	0	0	
1	17	0	
2	13	0	
3	25	0	
4	34	0	
5	37	12	~300Gb Traffic Transferred. Probe Started.
6	46	6	
7	46	8	
8	47	8	
9	40	10	
10	138	18	
11	178	36	
12	186	31	
13	140	23	
14	195	55	
15	119	15	
16	301	64	Traffic Increased Probes Increased
17	287	81	
18	281	35	
19	332	41	
20	13	4	Server Blocked. ~1950Gb Traffic Transferred.
21	12	0	
22	8	0	
23	3	0	
24	1	0	
25	0	0	

XRAY CONFIG
VLess-TCP-Reality
Port: 2053
Flow: None
SNI: ftp.debian.org:443

Server Status	BLOCKED
Runtime	24 Hours
Total Traffic	2 TB

Nginx Fallback	No
IR Domains & IPs	Blocked on Server



(Image 9 – AP1)
[Link]

Active-Probes

Test Type 2

The server, running VLess-TCP-Reality on Port 8080, operated for 75 hours, transferring ~8TB of data without being blocked. Legitimate traffic steadily increased, peaking at 343 SYNs by hour 68. Probes, likely from the Iranian firewall, began after transferring 1.2TB of data *(hour 24)* and spiked during hours 55 and 68, reflecting active targeting by censorship mechanisms.

Key Observations:

- Probes escalated alongside legitimate traffic, peaking at 343 SYNs *(hour 68)*, indicating persistent attempts to disrupt encrypted bypass mechanisms.
- Despite sustained probing and increased traffic, the server remained operational, demonstrating resilience against active filtering efforts.

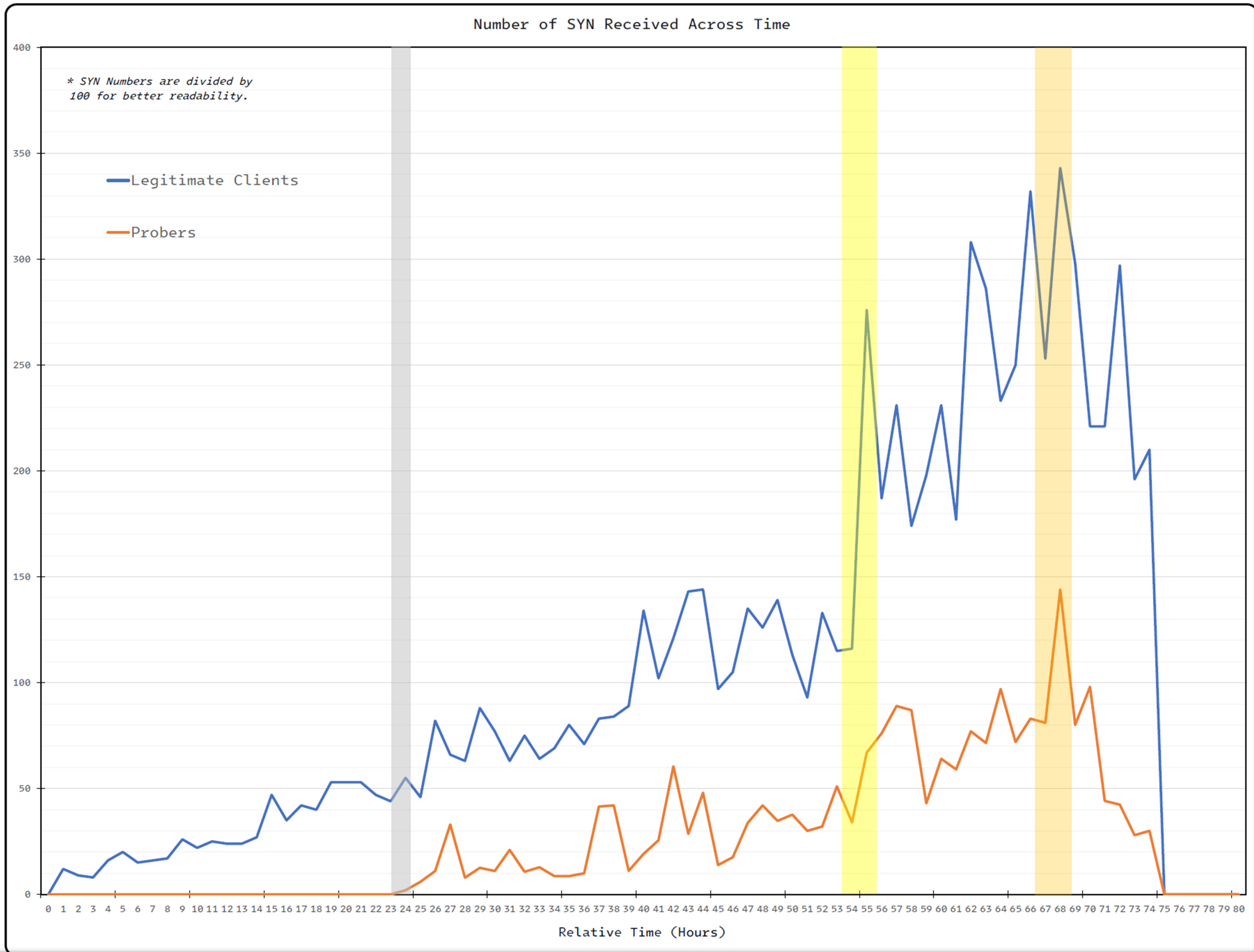
Number of SYN Received Across Time

Relative Time (Hours)	*Legitimate Clients*	*Probers*	
0	0	0	
1	12	0	
2	9	0	
3	8	0	
4	16	0	
5	20	0	
6	15	0	
7	16	0	
8	17	0	
9	26	0	
10	22	0	
11	25	0	
12	24	0	
13	24	0	
14	27	0	
15	47	0	
16	35	0	
17	42	0	
18	40	0	
19	53	0	
20	53	0	
21	53	0	
22	47	0	
23	44	0	
24	55	2	1.2 TB Traffic Transferred; Probes Started.
25	46	6	
26	82	11	
27	66	33	
28	63	8	
29	88	13	
30	77	11	
51	93	30	Redundent Data Hidden.
52	133	32	
53	115	51	
54	116	34	
55	276	67	Major Traffic increased; Probes increased.
56	187	76	
57	231	89	
58	174	87	
59	198	43	
60	231	64	
61	177	59	
62	308	77	
63	286	72	
64	233	97	
65	250	72	
66	332	83	
67	253	81	
68	343	144	Major Traffic increased; Probes increased.
69	298	80	
70	221	98	
71	221	44	
72	297	42	
73	196	28	
74	210	30	
75	0	0	Server Shutdown.
76	0	0	
77	0	0	
78	0	0	
79	0	0	
80	0	0	

XRAY CONFIG
VLess-TCP-Reality
Port: 8080
Flow: xtls-rprx-vision
SNI: ooklaserver :8080

Server Status	NOT BLOCKED
Runtime	75 Hours
Total Traffic	8 TB

Nginx Fallback	Yes
IR Domains & IPs	Blocked on Server



(Image 10 – AP2)
[Link]

Active-Probes

Test Type 3

The server, running VLess-TCP-Reality on Port 443, operated for 145 hours, transferring ~21TB of data without being blocked. Legitimate SYN requests steadily increased throughout the runtime, peaking at 1986 during hour 135. Probing activity, likely originating from the Iranian firewall, began after 4.2TB of data was transferred *(hour 61)* and intensified during three major spikes: hours 90, 117, and 132.

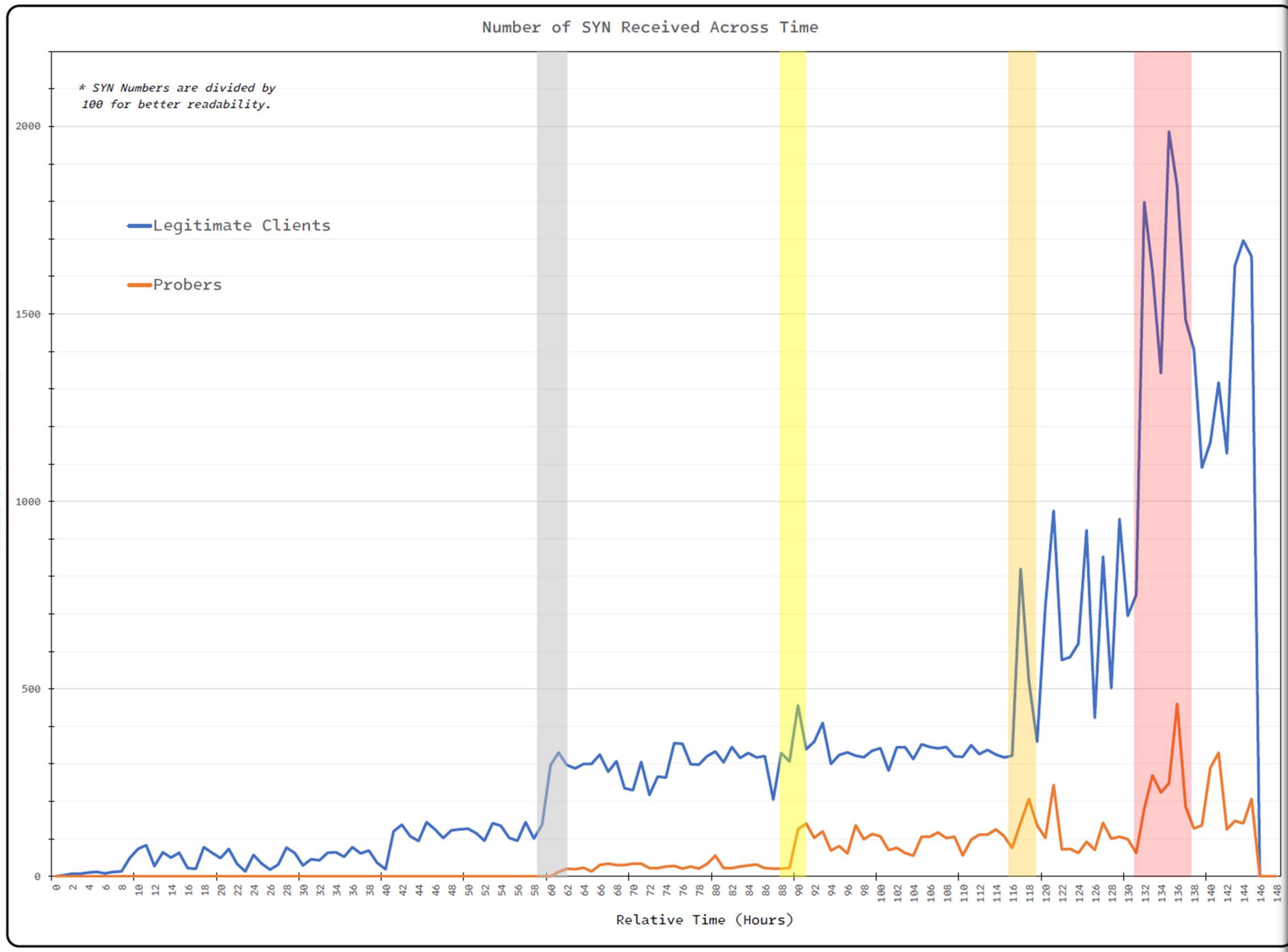
Key Observations:

- Probes began at hour 61 and grew significantly during major traffic surges. Probes peaked alongside legitimate traffic at 1986 SYNs during hour 135.
- Each major increase in legitimate traffic triggered a corresponding spike in probes, indicating systematic filtering efforts targeting high-traffic periods.
- Despite heavy traffic and persistent probing, the server remained operational, demonstrating robustness against the censorship mechanisms deployed.

Number of SYN Received Across Time

Relative Time (Hours)	*Legitimate Clients*	*Probers*	
0	0	0	
1	3	0	
2	7	0	
3	7	0	
4	10	0	
5	12	0	
6	8	0	
7	12	0	
8	13	0	
9	49	0	
10	73	0	
11	83	0	
12	27	0	
13	64	0	
14	50	0	
15	63	0	
16	22	0	
17	20	0	
18	78	0	
57	144	0	Redundent Data Hidden.
58	101	0	
59	138	0	
60	297	0	
61	330	12	4.2 TB Traffic Transferred; Probes Started.
62	297	20	
63	288	19	
64	300	23	
65	300	13	
66	325	30	
67	279	34	
86	321	22	Redundent Data Hidden.
87	205	21	
88	329	21	
89	306	22	
90	456	125	Major Traffic increased; Probes increased.
91	339	141	
92	359	103	
93	409	120	
94	300	69	
95	324	81	
96	331	61	
113	337	111	Redundent Data Hidden.
114	325	125	
115	317	107	
116	322	76	
117	820	143	Major Traffic increased; Probes increased.
118	520	206	
119	359	136	
120	720	103	
121	975	244	
122	577	72	
123	585	73	
124	620	62	
125	923	92	
126	423	71	
127	853	142	
128	502	100	
129	953	106	
130	695	99	
131	751	63	
132	1798	180	Major Traffic increased; Probes increased.
133	1613	269	
134	1342	224	
135	1986	248	
136	1840	460	
137	1483	185	
138	1405	128	
139	1090	136	
140	1158	290	
141	1317	329	
142	1128	125	
143	1628	148	
144	1696	141	
145	1653	207	
146	0	0	Server Shutdown.
147	0	0	
148	0	0	

XRAY CONFIG	
VLess-TCP-Reality	
Port: 443	
Flow: xtls-rprx-vision	
SNI: steal personal domain with *.com* tld	
Server Status	NOT BLOCKED
Runtime	145 Hours
Total Traffic	21 TB
Nginx Fallback	Yes
IR Domains & IPs	Blocked on Server



(Image 11 – AP3)
[Link]

Active-Probes

Summary

- ### Test Methodology

All Active Probe tests have been consolidated into three primary categories. The majority of these tests were conducted using Xray-core, supplemented by additional testing with various other cores and methods across Iran. The findings were consistent across both TLS and non-TLS protocols, indicating that the specific protocol used had minimal influence on the probing behaviour of the IRGFW. Notably, approximately 90% of our servers running TLS proxies and VPN tunnels were subject to probing by the IRGFW.

To manage these probes effectively, the Nginx webserver can be employed. It is important to note that probers should not be blocked outright; instead, they should be configured to receive neutral HTTP status codes *(e.g., 2XX, 3XX, 404, etc.)*.

- ### Probing Ratio

The average ratio of probers to legitimate SYN requests ranged from **0.2** to **0.3**. This meant that for every legitimate user, there were approximately **20%** to **30%** as many probes on average, indicating a relatively high level of active probing compared to legitimate traffic.

- ### Test Period & Relevance

Please note that these results reflect tests conducted up until September 2023 and are provided to showcase the active probing capabilities of the IRGFW at that time. As of December 2024, these findings are no longer applicable, as the IRGFW has since ceased using any active probing mechanisms.

The DPI
(Deep Packet Inspection)

The DPI

TLS Situation

The IRGFW consistently performs deep inspection and fingerprinting of TLS ClientHello handshakes to identify potential VPN or proxy traffic based on distinctive patterns, regardless of the TLS version used. While tools like uTLS can be employed to obscure some of these fingerprints, they do not fully eliminate detection, as uTLS itself has vulnerabilities that can still be identified by sophisticated DPI techniques.

We developed a series of tools to measure and analyze these behaviours. For instance, we set up a Nginx server hosting a standard website on a public *(whitelisted)* IP address. The site was accessible without issues across all major Iranian ISPs using Chrome and Firefox. However, when a DNS query *(using DoT or DoH)* was initiated from a popular DNS client on Windows, the TLS handshake failed to complete, resulting in a timeout.

When we tested with uTLS *(both official and fragmentation modes)*, the handshake was completed, indicating that the IRGFW had fingerprinted the DNS client. This issue also affects major VPN clients: despite having a whitelisted server IP and SNI domain, the TLS handshake times out.[32] However, when using a less common or non-standard client with different fingerprinting characteristics, the handshake succeeds, and the VPN tunnel is established without issues.

The DPI

IRGFW DPI consists of two central systems:

1. **The Active Part:** Check each international connection's first 1–17 kilobytes. This system looks for predefined signatures in the first packets of each stream, such as 0x16 0x3, which indicates a possible TLS type. It then looks for the SNI extension in this packet, which starts with 0x1 and includes the packet length. After identifying the SNI, it determines whether it is in the blocking hashtable. If the packet is not of the TLS type, the system looks for other signatures, such as SSH or HTTP. Regarding HTTP, the system looks for the Host header.

   Previously, the system was case-sensitive and sensitive to extra spaces, but it has been updated to eliminate all spaces. This active signature checking appears to be performed on specialized ASICs due to their high processing load, but even with powerful processors, delays and increased ping occur. People return home in the afternoon and activate their VPNs, causing the blocking system to become congested. It's worth noting that the operators in the active part differ, each having their own set of bugs, indicating that the system isn't wholly consistent.

2. **The Passive Part:** Before the recent update *(late Dec 2023 / early Jan 2024)*, the DPI system was fully active and could be deceived without causing any issues. However, after the update, MCI randomly samples some of each person's connections, passively capturing patterns of circumvention. These patterns include TLS in TLS, authentications, and standard VPN packet headers. For example, when using VLess *(V2ray/Xray)*, VLess sends a small authentication packet to each connection before sending the mainstream, ensuring the client is legitimate. Furthermore, when establishing a new VPN connection with another TLS connection, the passive blocking system searches for repeating patterns in small packets containing TLS or V2ray/Xray patterns. If the IP addresses and domains are discovered, they are flagged and reported to the blocking system every 4 hours *(time-pattern)*, where they are either throttled or blocked entirely.

The DPI

Possible Solution

To mitigate the risk of server blocking, the goal is to disrupt the patterns that enable detection. One way to do this is by modifying traffic patterns that are easily identifiable by servers. Injecting randomized packets at the start of each stream can help obscure the traffic's intent, making it harder for detection algorithms to classify it. Additionally, multiplexing multiple streams into fewer connections reduces the visibility of individual traffic flows, further decreasing the chance of detection.

For authentication traffic, injecting randomized packets and fragmenting them with varying padding and sizes can prevent the server from recognizing predictable patterns. By making the authentication process less uniform, you reduce the likelihood of it being flagged.

Blocking effectiveness relies on the inability to modify protocols or propagate changes to users easily. If users can adjust traffic patterns dynamically and apply these changes broadly, it undermines the server's ability to block based on fixed patterns. The ability to modify protocols *(such as through encryption, traffic obfuscation, or fragmentation)* helps maintain anonymity and reduce the risk of detection, making blocking attempts less effective.

This strategy hinges on continuous adaptation to avoid predictable behaviour that could be used for blocking or filtering.

Protocols Overview

Protocols Overview

These tests are conducted intensively with MahsaServer.com *(whenever possible)*; other tests were conducted anonymously in the real world and with Iranian users via the top five ISPs. The number of tests varies from 4 to 20 servers and tests for each protocol or method. The results are averaged, and the median of the results of all protocol tests. Also, all tests are conducted directly on a foreign server, and no middle or tunnelled servers are involved.

- **Socks5, SSTP, PPTP, IKEv2/IPsec:** Blocked by their fingerprints to all foreign IP addresses. **(Blacklist)**
- **L2TP:** Blocked. Many government officials use this protocol, but their Iranian IP addresses or IMEIs have been whitelisted. **(Blacklist)**
- **OpenVPN:** Completely blocked by its fingerprint in all major ISPs. **(Blacklist)**
    - **OpenVPN + Cloak:** Partially functional. Cloak was recently detected by IRGFW [29], resulting in minimal UL/DL speeds with high jitter. **(Graylist)**
- **Wireguard:** Completely blocked by all major ISPs but can function without limitations on some ISPs with a clean IP address and minimal traffic. Higher traffic leads to quick blocking.
    - **Obfuscated Wireguard:** As discussed in the UDP situation section, it can be used by modifying the handshake, but it's vulnerable to fingerprinting.
- **Shadowsocks** *(old and new encryptions and methods)*: Mostly blocked, occasionally graylisted. Some modifications allow connectivity but with high packet loss and jitter. **(Graylist)**
    - **ShadowSocks + Cloak:** Partially functional. Detected by IRGFW with minimal UL/DL speeds and high jitter **(Graylist).**
- **MTProto:** Mostly graylisted. When functional, it follows a strict time-pattern, leading to IP blockage within four days, but it can be extended to 2 weeks or more.
- **SoftEther:** Similar to Wireguard. Blacklisted by fingerprint and follows a strict time-pattern. **(Blacklist)**
- **SSH:** Partially functional on some ISPs and Gray-listed on others. Often follows a loose time-pattern. **(Graylist)**
    - **SSH-over-TLS:** Partially functional and often follows a loose time-pattern. **(Graylist)**

Protocols Overview

- **V2Ray/XRay/SingBox** *(v5.22.0/v24.12.18/v1.10.5)*:
  - **VMess-(TCP/WS/HU/GRPC)-NonTLS:** Works with a clean IP *(MCI and TCI firewalls only)* but is usually blocked within four days and up to two weeks in some cases.
  - **(VLess/VMess)-(TCP/WS/HU/GRPC/H2)-TLS:** Works with a clean IP but is often blocked within two weeks *(time-pattern).*
  - **REALITY/ShadowTLSv3:** Mostly blocked within four days *(sometimes within 24 hours)* unless used with a whitelisted SNI but usually blocked within two weeks, even with a whitelisted SNI. This behaviour strongly suggests that the IRGFW employs a reverse DNS mapping system to identify and block these types of protocols and destination IP addresses.
  - **Trojan:** Similar to V2Ray/Xray with TLS. Graylisted and follows a time-pattern.
- **Hysteria2:** Requires a QUIC-enabled destination IP *(Page 8 - UDP section).*
  - **Hysteria2 + Obfs** *(Salamander)*: QUIC may be completely disabled to some IPs, but Salamander Obfs can sometimes bypass this restriction if UDP works appropriately.
- **TUIC/JUICITY:** Similar to plain Hysteria2. Gray-listed with limited UL/DL bandwidth and high jitter.
- **Obfs4** *(for any protocols like OpenVPN/ShadowSocks/Tor)*: Mostly blocked but can work on some ISPs. Gray-listed and has exceptionally high jitter and UL limitations.
- **TOR** *(with every bridge combination)*: Mostly blocked. And rarely gray-listed with a limited speed.

Protocols Overview

- **CDN (Content Delivery Network):**

  Certain Content Delivery Networks *(CDNs)*, such as Cloudflare, are compatible with specific protocols that enhance security and privacy. A common configuration is VLess+(WS/gRPC)+TLS, which works effectively to conceal a Virtual Private Server *(VPS)* IP address by routing traffic through a CDN. This setup takes advantage of the CDN to obfuscate the source server's IP, making it harder for adversaries like the IRGFW to directly target the VPS.

  However, the SNI/Host field in the protocol configuration often serves as a vulnerability. When this field is located, the IRGFW can block it, effectively neutralizing the traffic. To mitigate this, fragmentation techniques are employed. Fragmentation involves splitting the SNI/Host domain into smaller components to prevent the firewall from reading or interpreting it properly. This method aims to outsmart the filtering mechanisms.[30]

  Despite these efforts, there are limitations. The IRGFW may escalate its countermeasures by blocking all connections to certain CDNs that are unable to interpret fragmented SNI/Host data. Furthermore, as of November 2024, Cloudflare appears to have implemented stricter security measures aimed at filtering out “bot-like” traffic. Unfortunately, traffic generated by tools such as V2ray/Xray is classified as bot traffic under these guidelines, leading to connection interruptions or outright blocking.

- **ECH/ESNI:**

  ECH, formerly known as ESNI, serves a similar purpose as fragmentation: preventing firewalls from reading the SNI domain. By encrypting the handshake process, ECH ensures that the SNI remains hidden from middleboxes and censorship mechanisms. This encryption disrupts the IRGFW’s ability to inspect the unencrypted handshake, effectively thwarting many censorship attempts.

  Historically, ECH and its predecessor ESNI faced outright blocking in countries with stringent censorship policies, such as Iran and China. However, in recent years, Iran has allowed the use of ECH, providing a potential avenue for bypassing restrictions. This is in contrast to China, where ECH and ESNI continue to be actively blocked by the Great Firewall *(GFW)*.[31]

  While ECH offers robust protection by encrypting the SNI, it remains vulnerable to infrastructure-level blocks. As noted in the CDN section, if the underlying network infrastructure *(e.g., IRGFW or Cloudflare)* decides to block certain types of encrypted traffic, ECH configurations can become ineffective. This vulnerability highlights the ongoing arms race between censorship circumvention techniques and the countermeasures deployed by oppressive regimes.

November 2024 Update

Update on the IRGFW

As of December 2024 *(and at the time of writing this report)*, the IRGFW has significantly scaled back its DPI functions. This reduction has led to the deactivation or minimal enforcement of previously rigorous blocking rules, time-based restriction patterns, and active probing protocols that formed the core of IRGFW's stringent internet control.

Currently, the primary ISP firewalls remain operational; however, they function with reduced thresholds, allowing only basic filtering without the in-depth traffic inspection and monitoring that DPI typically provides. Consequently, many protocols, such as VPNs, encrypted connections, and various UDP-based services that would normally face high rates of throttling, blocking, or graylisting, are experiencing fewer restrictions and lower instances of disruption. The current state reflects a temporary easing of censorship measures, as IRGFW's normally advanced DPI capabilities *(like detecting and fingerprinting traffic patterns, active packet sampling, and blocking via synchronized blacklists)* are not being actively applied.

This reduced control intensity may allow for increased data flow and somewhat more open access to previously restricted internet services. However, this shift may be reversible depending on future policy decisions and technological adjustments. While this shift may be temporary, it represents a notable pause in IRGFW's otherwise pervasive control measures, allowing for a brief window of increased connectivity and reduced censorship across Iran's internet landscape.

Last Words

Censorship and circumvention engage in a dynamic and relentless battle. Circumvention methods are continuously developed, deployed, and refined, only to be identified, disrupted, and neutralized by increasingly sophisticated filtering systems. In response, new strategies emerge, temporarily restoring access and perpetuating this endless cycle of adaptation and counter-adaptation.

It's crucial to recognize that the current reduction in filtering intensity by the Islamic Republic of Iran is not a permanent shift or a sign of leniency. Instead, it is a calculated pause, likely designed to provide time for the IRGFW and its associated systems to train and evolve. These systems are being fine-tuned to better detect and counteract new circumvention methods, preparing for a stricter and more effective resurgence. Such measures will enable tighter control during politically or socially critical periods when managing the flow of information is essential for maintaining authority.

In this environment, relying on a single method of circumvention is not just ineffective—it's dangerous. A sustainable approach demands a diverse toolkit of techniques, used in parallel. Employing multiple methods simultaneously—ranging from different protocols and encrypted channels to traffic obfuscation and fragmentation—greatly reduces the risk of complete disruption. Redundancy ensures that if one method is compromised, others remain functional, maintaining connectivity and access.

Ultimately, adaptability and strategic diversification are essential to counter increasingly advanced censorship mechanisms. Success in this battle requires constant innovation, proactive thinking, and the deployment of a wide range of tools to stay ahead of oppressive systems that continue to evolve. The fight for digital freedom is not a static challenge; it demands resilience, creativity, and a readiness to meet each new restriction with stronger, more agile solutions.

References

1. https://www.amnesty.org/en/latest/news/2023/09/what-happened-to-mahsa-zhina-amini/
2. https://www.ohchr.org/en/press-releases/2023/09/iran-one-year-anniversary-jina-mahsa-aminis-death-custody-heightened
3. https://github.com/net4people/bbs/issues/125
4. https://gfw.report
5. https://github.com/net4people/bbs/issues/129
6. https://apps.dtic.mil/sti/citations/AD1107324
7. https://dig.watch/updates/iran-to-implement-national-information-network-to-keep-people-off-the-internet
8. https://www.science.org/content/article/iran-s-researchers-increasingly-isolated-government-prepares-wall-internet
9. https://www.en-hrana.org/tag/tarh-e-sianat/
10. https://bgp.tools/rankings/IR?sort=cone
11. https://internetabad.factnameh.com/fa
12. https://mci.ir/introduction
13. https://irancell.ir/p/4471/
14. https://www.tci.ir/%D8%AA%D8%A7%D8%B1%DB%8C%D8%AE%DA%86%D9%87/
15. https://www.tic.ir/fa/introduce
16. http://www.iranet.ir/
17. https://www.ipm.ir/
18. https://www.paloaltonetworks.com/blog/2014/06/udp-malware-hiding-place-of-choice/
19. https://knowledgebase.paloaltonetworks.com/KCSArticleDetail?id=kA10g000000Clc6CAC
20. https://github.com/net4people/bbs/issues/140
21. https://ieeexplore.ieee.org/document/1404672
22. https://etchamber.ir/internet-report-2
23. https://github.com/net4people/bbs/issues/113
24. https://irgfw.report/blog/post1/
25. https://github.com/net4people/bbs/issues/224#issuecomment-1462268182
26. https://t.me/irgfw/6
27. https://github.com/XTLS/Xray-core/issues/2778
28. https://gfw.report/blog/gfw_shadowsocks/
29. https://github.com/net4people/bbs/issues/327
30. https://github.com/GFW-knocker/gfw_resist_tls_proxy
31. https://github.com/net4people/bbs/issues/43
32. https://github.com/net4people/bbs/issues/153

Technical Analysis of the IRGFW

Understanding The Iranian Great Firewall

Report 1

December 2024

IRGFW

irgfw.report