

# QUICsand: Quantifying QUIC Reconnaissance Scans and DoS Flooding Events

Marcin Nawrocki  
marcin.nawrocki@fu-berlin.de  
Freie Universität Berlin  
Germany

Thomas C. Schmidt  
t.schmidt@haw-hamburg.de  
HAW Hamburg  
Germany

Raphael Hiesgen  
Raphael.Hiesgen@haw-hamburg.de  
HAW Hamburg  
Germany

Matthias Wählisch  
m.waehlisch@fu-berlin.de  
Freie Universität Berlin  
Germany

## ABSTRACT

In this paper, we present first measurements of Internet background radiation originating from the emerging transport protocol QUIC. Our analysis is based on the UCSD network telescope, correlated with active measurements. We find that research projects dominate the QUIC scanning ecosystem but also discover traffic from non-benign sources. We argue that although QUIC has been carefully designed to restrict reflective amplification attacks, the QUIC handshake is prone to resource exhaustion attacks, similar to TCP SYN floods. We confirm this conjecture by showing how this attack vector is already exploited in multi-vector attacks: On average, the Internet is exposed to four QUIC floods per hour and half of these attacks occur concurrently with other common attack types such as TCP/ICMP floods.

## CCS CONCEPTS

• Security and privacy → Denial-of-service attacks; • Networks → Transport protocols; Public Internet.

### ACM Reference Format:

Marcin Nawrocki, Raphael Hiesgen, Thomas C. Schmidt, and Matthias Wählisch. 2021. QUICsand: Quantifying QUIC Reconnaissance Scans and DoS Flooding Events. In *ACM Internet Measurement Conference (IMC '21)*, November 2–4, 2021, Virtual Event, USA. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3487552.3487840>

## 1 INTRODUCTION

QUIC is a secure transport protocol originally developed by Google and tested in Chrome browsers since 2013 [26]. It has been recently standardized by the IETF as RFC 9000 [22] and at the same time enjoys rapidly growing deployment by major Web operators and browsers. In 2017, Google estimated that QUIC accounted for

7% of Internet traffic [26] and, by the end of 2020, Facebook announced that 75% of its Internet traffic is QUIC [14]. Despite its recent standardization, QUIC has already many implementations [37] and concurrently supported QUIC versions [40]. In 2021, scans of the complete IPv4 address space detected around 2 million QUIC servers [40].

Key design objectives in QUIC were privacy and security. QUIC was built to reduce the attack surface on the transport layer, which includes attacks such as reflective amplifications [41] and resource exhaustions. Security considerations in the QUIC RFC [22] span 18 pages and discuss properties against active and passive attackers.

In this paper, we report about early observations that indicate regularly ongoing attacks based on QUIC. We argue that the strong security model in QUIC does not preclude misuse and measure clear signals of DDoS attacks in Internet background radiation. We confirm these results by correlating our observations with several complementary data sources. Our findings indicate that QUIC servers are indeed prone to resource exhaustion attacks and these flaws are currently exploited in multi-vector attacks. We believe that it will be crucial to monitor such attack attempts early in the QUIC deployment phase before they unfold their full potential.

The main contributions of our paper summarize as follows:

- (1) We present the first study on QUIC Internet background radiation as seen by a large network telescope.
- (2) We show a significant bias by research scanners but also detect scanning activity from non-benign sources.
- (3) Surprisingly, we find high-volume backscatter events suggesting that QUIC is used in multi-vector resource exhaustion attacks, targeting well-known companies.
- (4) We benchmark a popular web server implementation to test its DoS resiliency and validate our observations.
- (5) We show the efficacy of QUIC's built in defense mechanism with RETRY messages, which remains unused in the wild based on our measurements.

In the remainder of this paper, we present background and related work in Section 2. We outline the QUIC attack scenarios that we base on in Section 3, and introduce our measurement method and data sources in Section 4. We analyse QUIC scanning and backscatter events in Section 5. Finally, we discuss our findings in Section 6.

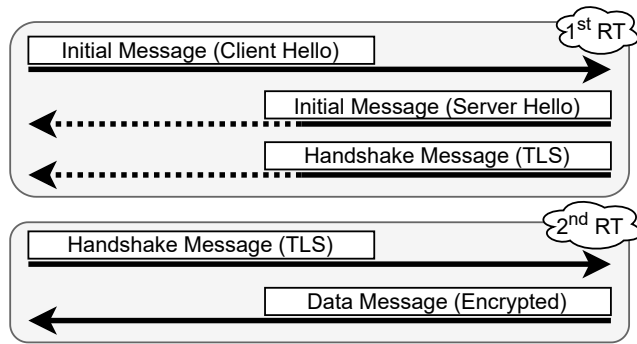
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

IMC '21, November 2–4, 2021, Virtual Event, USA

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-9129-0/21/11...\$15.00

<https://doi.org/10.1145/3487552.3487840>



**Figure 1: QUIC clients receive data with the second round-trip (RT). At the first RT, servers respond to unverified client IP addresses, which can be misused.**

## 2 BACKGROUND AND RELATED WORK

One of the major design goals in QUIC is the decrease of latencies for client-server applications by reducing round-trip times (RTTs) due to multiple, independent handshakes [10]. To this end, sequential handshakes from TCP, TLS, and HTTP have been merged into a single, comprehensive handshake process. Furthermore, to overcome delay because of TCP head-of-line blocking, QUIC is based on UDP [26].

**QUIC handshakes.** QUIC utilizes various handshake procedures to set up and resume connections. In the best case, when cached information of a prior connection is available to the client, encrypted application data can be sent immediately leading to a 0-RTT handshake. In the worst case, the handshake requires 3 RTTs [40]: If the client offers unsupported QUIC versions, the server first enforces a version negotiation and then proceeds with a *typical* [16] handshake.

A typical handshake is depicted in Figure 1. First, the client sends an Initial message including a TLS Client Hello, which is answered with an Initial message including a TLS Server Hello. This message is immediately followed by a handshake message containing the rest of the TLS server information (e.g., certificates). The setup is complete with a Handshake message from the client and is usually accompanied by a data request. Since the client is allowed to send data requests after the first round-trip, this handshake is described as a 1-RTT handshake. During the handshake, the server and client agree on a (i) source connection identifier (SCID) and a (ii) destination connection identifier (DCID), respectively, which can be used to identify the QUIC connection independently of the traditionally used connection 5-tuple.

**RETRY to mitigate resource exhaustion.** QUIC traffic is almost entirely encrypted with TLS 1.3 to prevent the ossification that middleboxes cause on protocols like TCP [19]. During the first RTT, a server responds to an unverified client IP address. This means that the server performs cryptographic operations for a potentially spoofed client. QUIC supports RETRY messages [22] to limit the attack surface of resource exhaustion attacks. RETRYs precede a typical QUIC handshake and force the client to respond with a unique token, which proves its authenticity. This mitigation,

however, adds a complete RTT, which conflicts with QUIC original design goals. Recent QUIC server implementations such as NGINX or Picoquic support RETRY [20] but based on the backscatter we observe RETRY seems rarely deployed.

**Related work.** Prior work focusses on three aspects. First, the performance benefits of QUIC, especially in low bandwidth, high latency, and high loss [3, 6, 9, 24] or multi-hop [8] scenarios. Second, the adoption of QUIC [29, 40, 43, 45]. Third, protocol security. Previous results [28] suggest that QUIC’s security weaknesses, such as insufficient forward secrecy or susceptibility to replay attacks, are introduced by the mechanisms used to reduce latency. To the best of our knowledge, this is the first paper that analyzes QUIC background radiation and reveals QUIC DoS traffic, showing that we still face a trade-off between small latencies and robust security guarantees in the wild.

## 3 QUIC ATTACK SCENARIOS

In this section, we briefly discuss those attacks that are most relevant in the context of this paper. An attacker has two common options [31] to exploit weaknesses in QUIC [7, 36]. First, an attacker could initiate state-overflow attacks to harm QUIC servers. Second, an attacker could trigger reflective amplification attacks to harm the network.

**State-overflow attacks.** To launch a state-overflow attack, an attacker would act as a QUIC client and induce connection states at a QUIC server, by initiating full handshakes. Any QUIC server answers to the connecting client with a unique Source Connection ID (SCIDs) and its TLS certificate. This part introduces cryptographic load on the server and forces the allocation of resources to maintain states. It is noteworthy that during the first round-trip (RT) of the full handshake the client is still unverified, which limits protection mechanisms in QUIC. To artificially increase the number of states, an attacker would randomly spoof the source IP addresses, source ports, or SCIDs. Floods benefit from spoofed IP addresses in particular as the backtracking of spoofed traffic is challenging [12, 15, 27, 34].

This attack is very similar to TCP SYN floods [11], which exploit the fact that the network stack needs to maintain all currently active connections. Spoofed connections fill up the connection queues at the victim and cause the rejection of legitimate TCP requests.

A large content network has started to mitigate QUIC floods [7] but the extent of this attack in the wild has not been studied yet.

**Reflective amplification attacks.** Since QUIC is based on UDP an attacker could easily launch a reflective amplification attack. An attacker would send an Initial packet including a spoofed source IP address to a QUIC server. The server then replies with an Initial QUIC message and a TLS handshake. The TLS handshake is larger compared to the client’s request, since it includes the server certificates. As long as the client is unverified (i.e., the client did not yet send messages that include information supplied by the server, see the QUIC RFC [22]) QUIC servers are only allowed to triple the bytes of a request in their response, though.

To increase the absolute number of bytes sent to the (spoofed) victim, an attacker would add padding bytes to the Initial message. Such strategy would not be suspicious. Sending large initial

handshake packets is suggested in order to allow the server to accommodate full certificates in a single message and thus reducing delays [30].

**Why amplification attacks are unlikely.** QUIC has been designed based on experiences with amplification attacks. Thus, it limits the size of replies to unverified clients to a factor of  $3\times$ . The major reason why QUIC is unlikely to be used for amplification attacks is the wide presence of other protocols that support much higher amplification factors (e.g., NTP  $500\times$  and DNS  $60\times$  [39]). Given TLS certificate compression [17], only 1%-9% of the server replies do not fit in a single message, depending on the initial client message size [30]. Furthermore, attackers tend to reuse their existing attack infrastructure and adapt very slowly to new protocols to conduct reflective amplification attacks [25]. As QUIC amplification attacks are currently unlikely, we focus on state-overflow attacks.

## 4 MEASUREMENT METHOD AND SETUP

We analyze QUIC background radiation, and in particular backscatter traffic, *i.e.*, responses to spoofed packets.

### 4.1 Method

Our vantage point is a network telescope, since network telescopes passively collect unsolicited traffic known as Internet background radiation (IBR). IBR consists of traffic resulting from research and malicious scans [33, 38], misconfigurations and bugs but also from responses sent by the victims of randomly spoofed, state-building attacks. Network telescopes have been used reliably to quantify DoS victims for more than 15 years [23, 32].

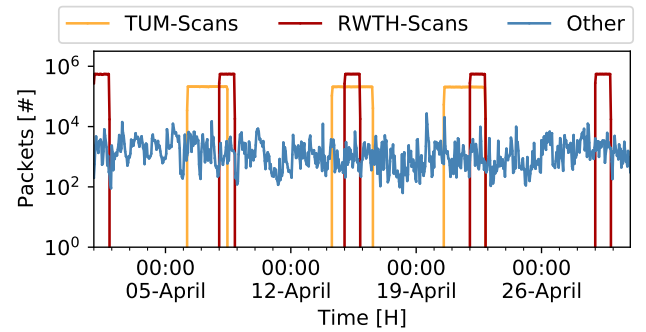
We identify QUIC traffic based on transport layer properties by selecting all UDP packets with a source or destination port UDP/443. This port-based classification has been proven sufficient in prior work [40]. To exclude false positives, we extend this common classification method by utilizing Wireshark payload dissectors, which also have been shown to be efficient for classification [35]. We verified QUIC detection by manually checking QUIC connections to well-known QUIC servers. All packets were correctly identified and dissected by Wireshark.

To group packets sent to the telescope into backscatter and scanning events, we mark all QUIC packets with source port UDP/443 as responses (*i.e.*, backscatter) and all packets with destination port UDP/443 as requests (*i.e.*, scans). These two sets are disjoint, as we do not find any packet with destination and source port set to UDP/443.

To find DoS events, we apply the notion of traffic sessions and DoS thresholds as defined by Moore *et al.* [32]. We acknowledge that these thresholds have been defined with the help of older traffic patterns, however, they are still used in recent work [23]. Along the line of our analysis we will learn that these thresholds (e.g., the *timeout* parameter) are still appropriate for current traffic patterns. For details on the impact of the threshold configurations, see Appendix B.

### 4.2 Data Sources

We utilize the UCSD Network Telescope [5] to observe both QUIC IBR and common TCP/ICMP IBR. This telescope is operated by the University of California San Diego and represents a /9 network prefix.,



**Figure 2: QUIC traffic seen at the UCSD network telescope. In the remaining analyses, we identify and remove the extreme bias of research scanners.**

*i.e.*, 1/512 of the available IPv4 address space. We are thus able to capture at least 2 % of any horizontal scan or randomly spoofed attack. In this paper, we focus on one month, April 1-30, 2021. Overall, we find 92 million QUIC packets during our measurement period.

In order to bolster our results and put them into context, we correlate our observations with several complementary data sources. We reuse data from active scans that explore QUIC servers [40], correlate IP addresses with the GreyNoise Honeypot Platform, and use meta data from PeeringDB. The active scans provide a set of potential victims, GreyNoise helps to assess multi-vector attacks based on advanced threat intelligence, and PeeringDB provides additional network descriptions. We mention that all data sets are in sync, spanning the same period of time.

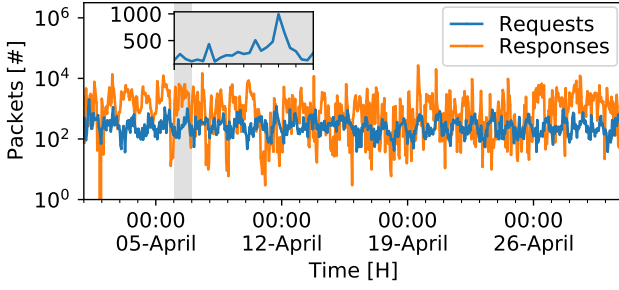
## 5 ANALYSIS

We now analyze QUIC IBR traffic. First, we show an overview of all IBR traffic and then focus on high-volume backscatter events.

### 5.1 Overview of QUIC IBR Traffic

**QUIC IBR is dominated by research scanners.** We first inspect the total packet count. Since multiple QUIC messages can be embedded in a single IP/UDP packet, we emphasize that we count packets and not individual QUIC messages. Overall, we observe 92 million QUIC packets in April 1-30, 2021. This data set is dominated by periodic scans that target the complete IPv4 address space. Each Internet-wide, single-packet scan sends  $2^{23} \sim 8 \times 10^6$  packets to the telescope. In total, 98.5% of QUIC packets are generated by research projects from two universities, Technische Universität München (TUM) and RWTH Aachen Universität (RWTH). We show the amount of packets per hour and compare research scanners with other traffic sources in Figure 2. Since these scanners clearly perform regular scans of the entire address space, we expect this bias also at other vantage points. We remove traffic from research scanners in the subsequent analyses to focus on the *other* traffic.

**QUIC requests follow diurnal patterns, responses are erratic.** In the sanitized traffic, we find 15% QUIC requests and 85% QUIC responses. Figure 3 shows the number of requests and responses per hour. Requests follow stable, diurnal patterns with peaks at



**Figure 3: Number of QUIC packets by type. Requests exhibit rather stable, diurnal activities with peaks at 6:00am and 6:00pm UTC (see insert for representative day). Responses are very erratic, hinting at flood events.**

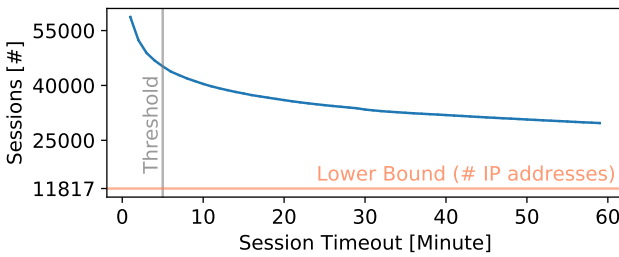
6:00am and 6:00pm UTC. We demonstrate this with the inlet for a representative day, April 06, 2021. Response traffic is very erratic, exhibiting high peaks and drops per event. This behavior might hint at DoS events [2, 4], which we will inspect in more detail in Section 5.2.

**We can reuse established thresholds for QUIC sessions.** In order to move from a packet-based to an event-based perspective, we now group singular packets into sessions. To this end, we aggregate packets using the source IP address and a timeout threshold, *i.e.*, packets from a specific source belong to a single session as long as the inactivity period between them is no longer than the timeout.

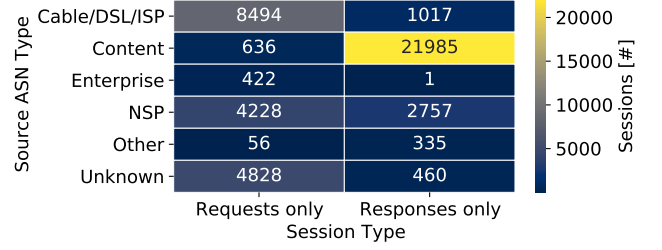
Figure 4 exhibits the number of detected sessions given a timeout value between 1 and 60 minutes. The lower bound of this plot is defined by  $\text{timeout}=\infty$ , which groups all packets of a source into a single event. We see a significant reduction of sessions until  $\sim 5$  minutes, which is why we choose this knee point as our threshold. This timeout is coherent with prior work that applied timeouts to IBR traffic [23, 32].

## 5.2 QUIC DoS Traffic

**Request sessions are non-benign and originate from eyeball networks. Response sessions are DoS backscatter from content providers.** We now inspect and contextualize the observed



**Figure 4: Influence of the timeout parameter on the number of sessions. We select 5 minutes as the final threshold to group correlated packets into sessions.**



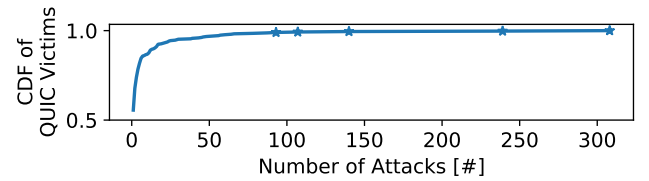
**Figure 5: Source network types of sessions. Requests originate predominantly from eyeballs. Responses are received almost exclusively from content networks.**

sessions. Overall, we find 18k sessions containing only requests, 26k sessions containing only responses. We do not observe sessions with both packet types. On average, a request session consists of 11 packets and response sessions of 44 packets.

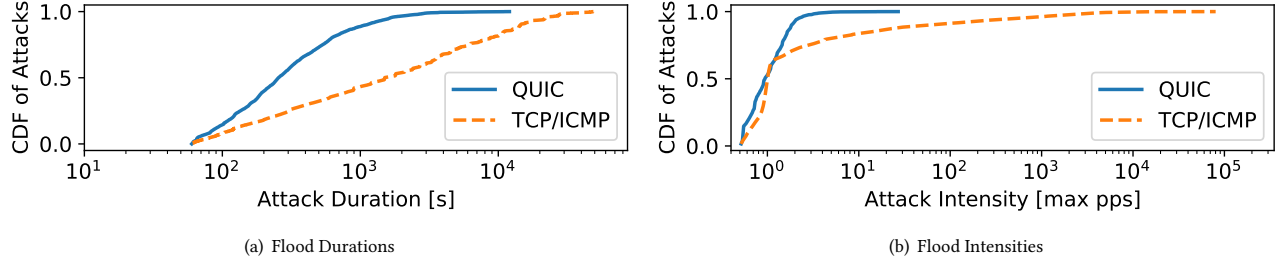
For each session, we map the source autonomous system number (ASN) and the AS network type using PeeringDB. We find that request sessions originate from eyeball networks and that response sessions are received from content provider networks, see Figure 5. This fits into the assumption that we (i) receive malicious scans from bots hosted in eyeball networks (*e.g.*, Mirai [1, 44]) and (ii) receive DoS backscatter from legitimate QUIC servers.

Taking a closer look at response sessions we find traffic that exhibits common DoS patterns, using, again, session timeouts and thresholds defined by Moore *et al.* [32]. To identify attacks, we select backscatter sessions with (i) more than 25 packets, (ii) a duration longer than 60 seconds, and (iii) a maximum packet rate of higher than 0.5pps, which is calculated over all 1-minutes slots of the respective event. Finally, we find 2905 attacks which correspond to 11% of all response sessions. Attacks target a total of 394 unique victims, with more than half being attacked only once, compare Figure 6. By correlating the victims with data obtained from active scans [40], we find that 98% of attacks target well-known QUIC servers. We take a closer look at the low-volume backscatter sessions and our threshold configuration in Appendix B.

To bolster our observations we correlate the sources of request sessions with data from a reactive vantage point, the GreyNoise honeypot platform. Using the GreyNoise classification, we do not find any signs of benign scanners and 2.3% of the sources are tagged



**Figure 6: CDF for number of attacks per QUIC flood victim. More than half of the victims are only attacked once during our measurement period. Last 5 data points are highlighted.**



**Figure 7: CDF of flood durations and intensities, comparing QUIC and TCP/ICMP. QUIC floods are shorter but the median intensity of QUIC floods is as severe as for common backscatter events.**

as known bruteforcers or part of a botnet such as Mirai or Eternalblue. Most request sessions originate from Bangladesh (34%), USA (27%), and Algeria (8%).

**QUIC floods are shorter but on average as severe as common TCP/ICMP floods.** We now compare QUIC DoS floods with floods for common protocols, *i.e.*, TCP and ICMP. The duration and intensity of attacks has been shown by Jonker *et al.* [23]. To allow for comparison, we reproduce the analysis based on our current setup. Overall, we find 282k attacks for common protocols.

Figure 7(a) shows the distribution of attack durations. QUIC floods are shorter than TCP/ICMP DoS attacks. The median QUIC flood lasts 255 seconds, for TCP/ICMP protocols we observe 1499 seconds. The median attack intensity, however, is close to 1 maximum packet per second (max pps) in both cases, see Figure 7(b). To estimate the traffic rate from the global Internet towards a victim, we may assume  $512 \times \text{max pps}$  since the UCSD telescope covers 1/512 of the total IPv4 address space.

**QUIC floods are part of multi-vector attacks and highly correlated with TCP/ICMP floods.** So far, we looked at QUIC floods in isolation. We now check whether they are part of a larger multi-vector attack towards a single victim. To our surprise, 51% of QUIC floods overlap in time with common (TCP/ICMP) DoS floods (see *concurrent* attacks in Figure 8). We require attacks to overlap for at least one second to label them as *concurrent*. Another 40% of QUIC floods target a victim in *sequence*, *i.e.*, the victim was also attacked by a TCP or ICMP flood during our measurement period but at a different time. In such cases, the gap between a QUIC and the nearest TCP/ICMP attack is 36 hours on average. More details about concurrent and sequential attacks, including an example, are presented in Appendix C. Only 9% of QUIC attacks do not relate to any TCP/ICMP event.

We argue that the reasons for multi-vector attacks are twofold. First, multi-vector attacks are harder to detect as they keep the traffic volumes for each attack vector low. Second, multi-vector attacks are more difficult to mitigate since they require more complex filter rules.

**Well-known content providers are under attack.** We find that 58% of attacks target Google and 25% target Facebook. Figure 9 compares attack properties for these two content providers to showcase potential differences in attack patterns across content providers and their deployed QUIC variants. We consider SCIDs in the backscatter

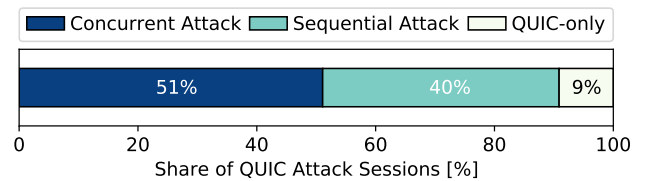
traffic. The SCID is a QUIC-specific feature, which may serve to assess allocated resources because a context is reserved for each unique connection. We do not show the number of DCIDs in this figure, since they are not available in our backscatter traffic and they are not required to route to the correct endpoint [22]. We carefully checked that the packets are valid, though, by verifying that the DCID length attribute is set to zero.

Overall, the number of spoofed client IP addresses is relatively low. The randomization of client ports, however, is the driving factor for new SCIDs at the attacked server. Despite a lower packet count, Google reacts with more SCIDs, which indicates a higher server load. We observe QUIC variant mvfst-draft-27 (95%) for Facebook attacks, and draft-29 (78%) for Google.

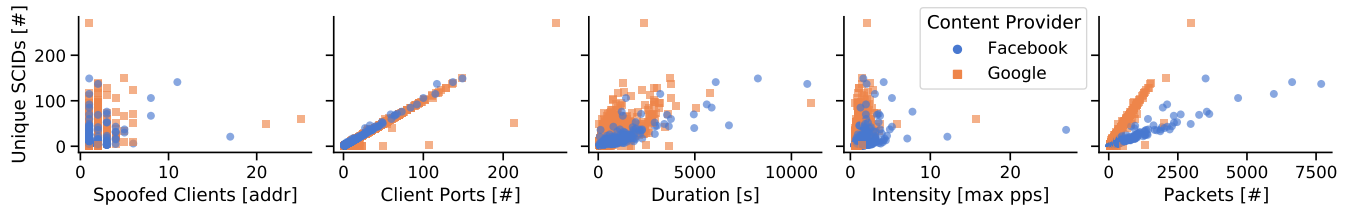
These results suggest that operators may protect against QUIC floods by filtering based on common transport protocol features (*i.e.*, ports) instead of using QUIC-specific features (*i.e.*, SCIDs), which eases the deployment of countermeasures.

## 6 DISCUSSION AND OUTLOOK

**Attack patterns are valid.** Captured QUIC events that are suspect to DoS consist of 31% Initial and 57% Handshake messages on average. The Initial messages we observe do not contain an (unencrypted) TLS Client Hello and thus can be attributed to (encrypted) Server Hello replies. Hence, these packets match the backscatter pattern of a QUIC attack (see Section 2). QUIC sends multiple UDP packets in response to the Initial packet: The first packet contains one Initial QUIC packet carrying the Server Hello and one encrypted Handshake message followed by a second datagram with a single Handshake message. The ratio



**Figure 8: Multi-vector attacks: Half of the QUIC attacks occur concurrently with TCP/ICMP floods.**



**Figure 9: >83% of attacks target two content providers. QUIC floods utilize multiple client addresses and ports. Despite a lower packet count, Google reacts with more SCIDs, indicating higher server load.**

in these packets roughly matches the ratio of one third Initial packets and two thirds Handshake messages.

**RETRY attack mitigation is not deployed.** We did not capture any RETRY messages. The absence of RETRY indicates the lack of deployment of a defense mechanism. To validate this observation, we select the ten most frequently attacked servers from Google and Facebook. When actively connecting to these servers with a QUIC client we also do not receive any RETRY messages. This supports the observations that we made based on data in the telescope.

Although the QUIC implementations of Google and Facebook support RETRY messages (Google since mid 2019 [18], Facebooks since the end of 2020 [21]), it looks like these content providers deliberately decide to not utilize this feature. This decision is potentially due to the performance penalty of RETRY messages. However, for frequently utilized services as in the case of large content providers, this penalty could be alleviated by the session resumption feature in QUIC. Also, RETRYs could be deployed adaptively and only used when high load occurs.

**QUIC servers quickly experience DoS.** To check whether the observed QUIC floods can be attributed to DoS attacks we experimentally analyzed the impact of QUIC handshakes on a common web server implementation. For this, we set up NGINX on a modern 128-core machine with 512 GB of RAM, which connects to a client via Gigabit Ethernet. NGINX supports the QUIC RFC [22] and eBPF optimizations [13]. We record 500,000 packets using the QUIC client quiche, version 0.9.0, Cloudflare’s QUIC reference implementation. To simulate attacks, we then replay *only* client Initial messages at varying packet rates towards new server instances. Replaying avoids bias from hand-crafting QUIC packets.

Our results are summarized in Table 1 alongside the configuration (we use 1024 client connections per worker which is twice the default). Since each request elicits four datagrams in response (two datagrams with Initial and Handshake packets plus two keep-alive PING packets after a short delay) we expect four times as many server responses. To determine how many requests were answered we match the respective DCIDs and SCIDs and calculate the service availability ratio.

With four worker processes even small packet rates can lead to significantly reduced service availability. At 1,000 probes per second (pps) only 7% of our requests are answered. In auto-mode, NGINX deploys 128 workers, but larger attack volumes of 10,000 pps continue to impact its availability. Note that these attacks do not impair the general availability of the machine but focus on the service. Extrapolating our observed 27 pps from a /9 IP prefix to

**Table 1: Tests on a local NGINX instance show that the backscatter volume we observed can significantly impact the responsiveness of the web server.**

Attack	NGINX Config		Results			
	Volume [pps]	QUIC Retry	Workers [#]	Client [# Req]	Server [# Resp]	Service Available RTT
10	✗	4	3,001	12,004	100%	✗
100	✗	4	30,001	81,680	68%	✗
1,000	✗	4	300,001	81,680	7%	✗
1,000	✗	auto=128	300,001	1,200,004	100%	✗
10,000	✗	auto=128	500,000	522,752	26%	✗
100,000	✗	auto=128	498,991	322,158	26%	✗
1,000	✓	4	300,001	300,001	100%	✓
10,000	✓	4	500,000	500,000	100%	✓
100,000	✓	4	500,000	500,000	100%	✓

the size of the Internet leads us to believe that attacks of more than 10,000 pps ( $27 \cdot 512 = 13,824$ ) are ongoing. In our benchmarks, RETRY packets successfully mitigate these attacks but they come at the high cost of an additional round trip.

The vulnerability based on the QUIC stateful handshake is not specific to an implementation, but relates to the protocol design. We expect all implementations to be prone to this attack type, the exact pps rates might vary, though. A recent benchmark of picoquic [20] also observed DoS at around 10,000 pps but a successful attack mitigation with RETRYs. Latest interoperability tests [42] show that the majority of QUIC server and client implementations correctly support the RETRY option. This leaves the decision on robustness versus speed up to the service providers.

**Attack duration and intensity.** We found that QUIC floods are shorter but the median max packet rate is similar compared to TCP and ICMP. The max packet rate is an indicator of the attack intensity but it also reflects the capability of a victim to sustain under load—for well-provisioned victims we likely observe higher rates as those victims are still able to send data. This also applies to the observed durations. Backscatter events can stop for various reasons: (i) the attack has ended, (ii) a mitigation was initiated, or (iii) the attacked service is completely unresponsive. Hence, shorter durations might indicate that QUIC attacks lead to a faster resource exhaustion compared to common protocols. Analyzing this, e.g., by using reactive scans or correlating with other data, will be part of our future work.

## ACKNOWLEDGMENTS

We would like to thank our shepherd Jelena Mirkovic and the anonymous reviewers for their helpful feedback. This work was partly supported by the German Federal Ministry of Education and Research (BMBF) within the project PRIMENet.

## REFERENCES

- [1] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztin, Jaime Cochran, Zakir Durumeric, J. Alex Halderman, Luca Invernizzi, Michalis Kallitsis, Deepak Kumar, Chaz Lever, Zane Ma, Joshua Mason, Damian Menscher, Chad Seaman, Nick Sullivan, Kurt Thomas, and Yi Zhou. 2017. Understanding the Mirai Botnet. In *26th USENIX Security Symposium (USENIX Security 17)*. USENIX Association, Vancouver, BC, 1093–1110.
- [2] Eray Balkanlı and A. Nur Zincir-Heywood. 2014. On the Analysis of Backscatter Traffic. In *Proc. of IEEE LCN*. IEEE, Piscataway, NJ, USA, 671–678.
- [3] Prasenjeet Biswal and Omprakash Gnawali. 2016. Does QUIC Make the Web Faster?. In *Proc. of IEEE Global Communications Conference (GLOBECOM)*. IEEE Press, Piscataway, NJ, USA, 6 pages.
- [4] Norbert Blenn, Vincent Ghiète, and Christian Doerr. 2017. Quantifying the Spectrum of Denial-of-Service Attacks through Internet Backscatter. In *Proc. of the 12th International Conference on Availability, Reliability and Security (Reggio Calabria, Italy) (ARES '17)*. ACM, New York, NY, USA, 10 pages.
- [5] CAIDA. 2012. The UCSD Network Telescope. [http://www.caida.org/projects/network\\_telescope/](http://www.caida.org/projects/network_telescope/) Last modified: January 29, 2018.
- [6] Gaetano Carlucci, Luca De Cicco, and Saverio Mascolo. 2015. HTTP over UDP. In *Proc. of the 30th Annual ACM Symposium on Applied Computing*. ACM, New York, NY, USA, 6 pages.
- [7] Cloudflare. [n.d.]. What is a QUIC flood DDoS attack? QUIC and UDP floods. Blog. <https://www.cloudflare.com/ko-kr/learning/ddos/what-is-a-quic-flood/> Last Access: May 2021.
- [8] Quentin De Coninck and Olivier Bonaventure. 2017. Multipath QUIC. In *Proc. of ACM CoNEXT*. ACM, New York, NY, USA, 160–166.
- [9] Sarah Cook, Bertrand Mathieu, Patrick Truong, and Isabelle Hamchaoui. 2017. QUIC: Better for what and for whom?. In *Proc. of IEEE International Conference on Communications (ICC)*. IEEE Press, Piscataway, NJ, USA, 6 pages.
- [10] Yong Cui, Tianxiang Li, Cong Liu, Xingwei Wang, and Mirja Kühlewind. 2017. Innovating Transport with QUIC: Design Approaches and Research Challenges. *IEEE Internet Computing* 21, 2 (March 2017), 72–76.
- [11] W. Eddy. 2007. *TCP SYN Flooding Attacks and Common Mitigations*. RFC 4987. IETF.
- [12] Jasper Eumann, Raphael Hiesgen, Thomas C. Schmidt, and Matthias Wählisch. 2019. *A Reproducibility Study of "IP Spoofing Detection in Inter-Domain Traffic"*. Technical Report arXiv:1911.05164. Open Archive: arXiv.org. <https://arxiv.org/abs/1911.05164>
- [13] Inc. F5. 2021. Our Roadmap for QUIC and HTTP/3 Support in NGINX. Website. <https://www.nginx.com/blog/our-roadmap-quic-http-3-support-nginx/> Last Access September 2021.
- [14] Facebook. 2020. How Facebook is bringing QUIC to billions. Engineering Blog. <https://engineering.fb.com/2020/10/21/networking-traffic/how-facebook-is-bringing-quic-to-billions/> Last Access May 2021.
- [15] Osvaldo Fonseca, Ítalo Cunha, Elverson Fazzion, Wagner Meira, Brivaldo Junior, Ronaldo A. Ferreira, and Ethan Katz-Bassett. 2020. Tracking Down Sources of Spoofed IP Packets. In *Proc. of IFIP Networking Conference*. IEEE Press, Piscataway, NJ, USA, 208–216.
- [16] Eva Gagliardi and Olivier Levillain. 2020. Analysis of QUIC Session Establishment and Its Implementations. In *Information Security Theory and Practice*. LNCS, Vol. 12024. Springer Nature, Switzerland, 169–184.
- [17] A. Ghedini and V. Vasiliev. 2020. *TLS Certificate Compression*. RFC 8879. IETF.
- [18] Google. [n.d.]. quiche. Github Repository. <https://github.com/google/quiche/> search?p=2&q=retry&type=commits Last Access: September 2021.
- [19] Michio Honda, Yoshifumi Nishida, Costin Raiciu, Adam Greenhalgh, Mark Handley, and Hideyuki Tokuda. 2011. Is it still possible to extend TCP?. In *Proc. of ACM IMC*. ACM, New York, NY, USA, 181–194. <https://doi.org/10.1145/2068816.2068834>
- [20] Christian Huitema. 2020. A simple test of DDOS attacks on QUIC. Blog. <https://huitema.wordpress.com/2020/09/22/a-simple-test-of-ddos-attacks-on-quic/> Last Access May 2021.
- [21] Facebook Incubator. [n.d.]. mvfst. Github Repository. <https://github.com/facebookincubator/mvfst/search?p=2&q=retry&type=commits> Last Access: September 2021.
- [22] J. Iyengar and M. Thomson. 2021. *QUIC: A UDP-Based Multiplexed and Secure Transport*. RFC 9000. IETF. <https://doi.org/10.17487/RFC9000>
- [23] Mattijs Jonker, Alistair King, Johannes Krupp, Christian Rossow, Anna Sperotto, and Alberto Dainotti. 2017. Millions of Targets under Attack: A Macroscopic Characterization of the DoS Ecosystem. In *Proc. of ACM IMC (London, United Kingdom)*. ACM, New York, NY, USA, 100–113. <https://doi.org/10.1145/3131365.3131383>
- [24] Arash Molavi Kakhki, Samuel Jero, David Choffnes, Cristina Nita-Rotaru, and Alan Mislove. 2017. Taking a long look at QUIC. An Approach for Rigorous Evaluation of Rapidly Evolving Transport Protocols. In *Proc. of ACM IMC*. ACM, New York, NY, USA, 290–303.
- [25] Daniel Kopp, Christoph Dietzel, and Oliver Hohlfeld. 2021. DDoS Never Dies? An IXP Perspective on DDoS Amplification Attacks. In *Passive and Active Measurement (LNCS, Vol. 12671)*. Springer Nature, Switzerland, 284–301.
- [26] Adam Langley, Alistair Riddoch, Alyssa Wilk, Antonio Vicente, Charles Krasnic, Dan Zhang, Fan Yang, Fedor Kouranov, Ian Swett, Janardhan Iyengar, Jeff Bailey, Jeremy Dorfman, Jim Roskind, Joanna Kulik, Patrik Westin, Raman Tenneti, Robbie Shade, Ryan Hamilton, Victor Vasiliev, Wan-Teh Chang, and Zhongyi Shi. 2017. The QUIC Transport Protocol. In *Proc. of ACM SIGCOMM*. ACM, New York, NY, USA, 183–196.
- [27] Franziska Lichtblau, Florian Streibelt, Thorben Krüger, Philipp Richter, and Anja Feldmann. 2017. Detection, Classification, and Analysis of Inter-Domain Traffic with Spoofed Source IP Addresses. In *Proc. of ACM IMC (London, United Kingdom)*. ACM, New York, NY, USA, 86–99.
- [28] Robert Lychev, Samuel Jero, Alexandra Boldyreva, and Cristina Nita-Rotaru. 2015. How Secure and Quick is QUIC? Provable Security and Performance Analyses. In *Proc. of IEEE Symposium on Security and Privacy*. IEEE Press, Piscataway, NJ, USA, 214–231.
- [29] Diego Madariaga, Lucas Torrealba, Javier Madariaga, Javiera Bermúdez, and Javier Bustos-Jiménez. 2020. Analyzing the Adoption of QUIC From a Mobile Development Perspective. In *Proc. of the SIGCOMM Workshop on the Evolution, Performance, and Interoperability of QUIC (EPIQ)*. ACM, New York, NY, USA, 35–41.
- [30] Patrick McManus. 2020. Does the QUIC handshake require compression to be fast? Fastly Blog. <https://www.fastly.com/blog/quic-handshake-tls-compression-certificates-extension-study> Last Access May 2021.
- [31] Jelena Mirkovic and Peter Reiher. 2004. A Taxonomy of DDoS Attack and DDoS Defense Mechanisms. *ACM Sigcomm Computer Communication Review* 34, 2 (April 2004), 39–53.
- [32] David Moore, Colleen Shannon, Douglas J. Brown, Geoffrey M. Voelker, and Stefan Savage. 2006. Inferring Internet Denial-of-Service Activity. *ACM Trans. Comput. Syst.* 24, 2 (May 2006), 115–139.
- [33] David Moore, Colleen Shannon, and k claffy. 2002. Code-Red: A Case Study on the Spread and Victims of an Internet Worm. In *Proc. of the 2nd ACM SIGCOMM Workshop on Internet Measurement*. ACM, New York, NY, USA, 273–284.
- [34] Lucas Müller, Matthew Luckie, Bradley Huffaker, Kc Claffy, and Marinho Barcellos. 2019. Challenges in Inferring Spoofed Traffic at IXPs. In *Proc. of ACM CoNEXT*. ACM, New York, NY, USA, 96–109.
- [35] Marcin Nawrocki, Thomas C. Schmidt, and Matthias Wählisch. 2021. Industrial Control Protocols in the Internet Core: Dismantling Operational Practices. *Wiley International Journal of Network Management* (2021). <https://doi.org/10.1002/nem.2158>
- [36] Nexusguard Research. 2020. Could QUIC turn into the next most prevalent amplification attack vector? Nexusguard Blog. <https://blog.nexusguard.com/could-quic-turn-into-the-next-most-prevalent-amplification-attack-vector> Last Access September 2021.
- [37] Maxime Piraux, Quentin De Coninck, and Olivier Bonaventure. 2018. Observing the Evolution of QUIC Implementations. In *Proc. of the Workshop on the Evolution, Performance, and Interoperability of QUIC (EPIQ)*. ACM, New York, NY, USA, 8–14.
- [38] Philipp Richter and Arthur Berger. 2019. Scanning the Scanners: Sensing the Internet from a Massively Distributed Network Telescope. In *Proc. of ACM IMC (Amsterdam, Netherlands)*. ACM, New York, NY, USA, 144–157.
- [39] Christian Rossow. 2014. Amplification Hell: Revisiting Network Protocols for DDoS Abuse. In *Proc. of NDSS*. Internet Society, 15 pages.
- [40] Jan Rüh, Ingmar Poesse, Christoph Dietzel, and Oliver Hohlfeld. 2018. A First Look at QUIC in the Wild. In *Passive and Active Measurement (LNCS, Vol. 10771)*. Springer Nature, Switzerland, 255–268.
- [41] Fabrice J. Ryba, Matthew Orlinski, Matthias Wählisch, Christian Rossow, and Thomas C. Schmidt. 2015. *Amplification and DRDoS Attack Defense – A Survey and New Perspectives*. Technical Report arXiv:1505.07892. Open Archive: arXiv.org. <http://arxiv.org/abs/1505.07892>
- [42] Marten Seeman. 2021. QUIC Interop Runner. Blog. <https://interop.seemann.io/?test=s> Last Access September 2021.
- [43] Jean-Pierre Smith, Prateek Mittal, and Adrian Perrig. 2021. Website Fingerprinting in the Age of QUIC. *Proc. Priv. Enhancing Technol.* 2021, 2 (2021), 48–69.
- [44] Avast Software s.r.o. 2020. The return of the Mirai botnet. <https://blog.avast.com/return-of-mirai-botnet-avast>
- [45] Martino Trevisan, Danilo Giordano, Idilio Drago, Maurizio Matteo Munafò, and Marco Mellia. 2020. Five Years at the Edge: Watching Internet From the ISP Network. *IEEE/ACM Transactions on Networking* 28, 2 (2020), 561–574.

## A ARTIFACTS

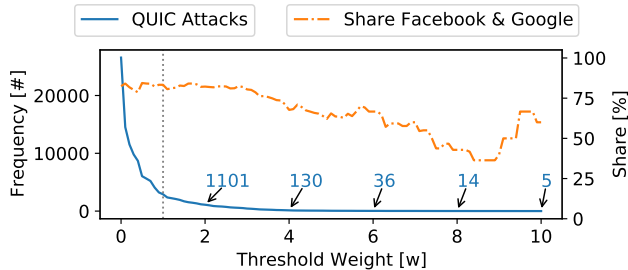
We support reproducible research. All artifacts of this paper are available on <https://doi.org/10.5281/zenodo.5504168>.

## B NON-ATTACK BACKSCATTER AND THRESHOLD CONFIGURATION

We reused thresholds defined by Moore *et al.* [32] to infer DoS events. We classified 11% of response sessions as attacks. Although this seems like an extreme reduction of events, we argue that underestimation is better than overestimation, because this prevents false positives. Nevertheless, we also checked the excluded response sessions for deviating trends. We do not find any anomalies. Excluded events have a median intensity of 0.18 max pps, a duration of 7 seconds, and consist of 11 packets. Such low-volume events point to misconfigurations and are most likely insignificant for our DoS analysis, hence, should be excluded.

To further understand the effects of our threshold configuration, we introduce the threshold weight  $w$ . We multiply each threshold by  $w$ , which leads to a more relaxed ( $w < 1$ ) or stricter ( $w > 1$ ) attack detection. If  $w = 1$ , the default threshold configuration as defined by Moore *et al.* [32] is used.

Figure 10 shows the number of detected attacks. We exclude many low-volume backscatter events for  $w \leq 0.3$ , but even for an extreme configuration of  $w = 10$  we still classify five backscatter sessions as attacks. On the secondary  $y$ -axis, we show that the share of well-known content providers remains high independently of  $w$ . These results bolster our main insight that QUIC Initial floods are used to attack large content providers.



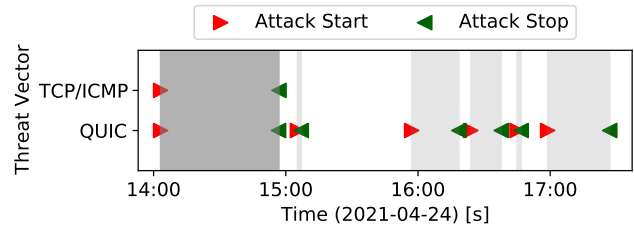
**Figure 10: Varying the DoS threshold defined by Moore *et al.* [32] to show the impact on the number of detected attacks and the relative share of affected content infrastructures. Even for a very strict threshold configuration of  $w = 10$ , we find QUIC attacks.**

## C DETAILS ABOUT ATTACKS

We introduced concurrent (*i.e.*, multi-vector) and sequential attacks in Section 5.2. In this section, we illustrate those attacks based on a concrete example and present more details about the time overlap between concurrent QUIC and TCP/ICMP attacks, as well as the time gap between sequential attacks.

### C.1 Illustration of Multi-vector versus Sequential Attacks

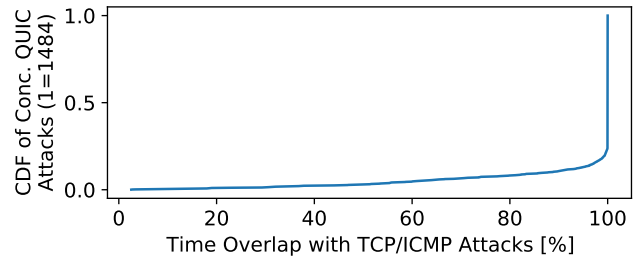
In Figure 11, we illustrate concurrent and sequential attacks based on a snapshot for a single victim. First, the victim is attacked by one QUIC and one TCP/ICMP attack that take place at the same time (or *concurrently*). Please note that these two attacks have an almost perfect overlap but we classify any two attacks as concurrent if the respective time ranges overlap in at least a single time unit, *i.e.*, they share at least one mutual second. Such a perfect overlap is very likely, compare Section C.2. We then observe five QUIC attacks in *sequence* to the first TCP/ICMP attack.



**Figure 11: Attacks towards a single victim. We observe one concurrent usage of attack vectors, *i.e.*, a multi-vector attack, followed by five sequential QUIC floods.**

### C.2 Overlap of Concurrent Attacks

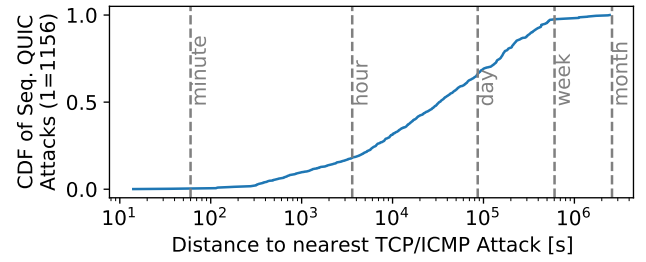
We now investigate how QUIC attacks overlap with common TCP or ICMP attacks. To this end, we calculate the share of overlapping seconds for each QUIC attack that is part of a concurrent attack. Figure 12 shows the distribution. We find a high correlation between QUIC and TCP/ICMP attacks. Three quarters of all concurrent QUIC attacks occur completely in parallel to an TCP/ICMP attack (100% in the CDF). On average, multi-vector QUIC attacks share 95% of the attack time with common attacks.



**Figure 12: Attack overlap of multi-vector attacks. Most concurrent QUIC attacks almost completely overlap with attacks that use common protocols.**

### C.3 Time Gaps Between Sequential QUIC Attacks and TCP/ICMP Attacks

We label an attack session *sequential attack* when we observe QUIC and TCP/ICMP attack traffic to same victim but QUIC and TCP/ICMP attacks do not overlap. Figure 13 exhibits the distribution of time gaps between both attack vectors. There is a break of more than one hour for 82% of the sequential attacks. In some cases, a break may take up to 28 days. These long time gaps suggest that sequential attacks are indeed not part of a multi-vector attack.



**Figure 13: Distribution of time gaps between the end (or start) of a sequential QUIC attack and the start (or end) of a TCP/ICMP attack.**