A Study of Deployed Defenses Against Reflected Amplification Attacks in QUIC

Aurélien Buchet, Cristel Pelsser
UCLouvain, Belgium
firstname.lastname@uclouvain.be

Abstract—**While the QUIC specification now includes mechanisms to prevent DoS attacks, they might not always be enforced by servers. With the increasing deployment of QUIC servers, it is now becoming more important to avoid vulnerabilities that could be exploited on a large scale. This paper presents an extensive study of the current state of QUIC servers and how they implement the mechanisms to prevent DoS attacks. The paper focuses on two different amplification DoS attacks that can be performed using QUIC HTTP/3 servers, enabled by the handshake and the connection migration mechanism. We investigate how QUIC servers respond to these attacks and if they are compliant with the general guidelines regarding the amplification protection. Our results show that while a large proportion of QUIC servers are respectful of the specification, around 20% of the IPv4 servers tested are still breaking the amplification limit for the handshake attack while most of the IPv6 servers are compliant. Most of the servers who support connection migration use the path validation mechanism, preventing the attack on connection migration. Overall, the amplification factor of the attacks remains quite low with a median slightly lower than the limit of 3, set in the standard, for the handshake attack and under 1 for the migration attack.**

I. Introduction

Distributed Denial of service attacks are ever increasing in number [32] and volume with Clouflare facing a record 3.8 Tbps [1] in September 2024. These create a significant burden on CDNs (among other players) that need to absorb the rise of traffic and mitigate the attacks. Furthermore, new such attacks keep being discovered [22], [5]. These attacks rely on spoofing the IP address of the victim to redirect large volumes of data to it. This attack method, despite being well known, is widely used because the classic countermeasure, Source Address Validation (SAV), is still not deployed on all networks [18], [23]. SAV consists of verifying that the traffic going through a network has a source IP address that is allowed to enter or leave the network. This verification is done at the edge of the network and the rules to prevent IP address spoofing have been standardized by the IETF [28], [2]. SAV provides the best protection when applied in stub networks, at the edge of the Internet, where it is easy to determine if the source IP is legitimate. As stubs compose the majority of the ASs in the Internet, a lot of players need to enforce SAV to prevent most DDoS attacks. Since this is not likely to happen in a near future, it is important to test the different Internet protocol abilities to protect themselves against amplification abuse. This paper focuses on DoS attacks that rely on the QUIC protocol [14].

The QUIC protocol has been quite popular since its introduction by Google in 2013 and has since been developed by many actors from the industry and academia. It is now standardized by the IETF and supported by most major browsers and content providers. QUIC was designed to replace TCP [10] for web traffic, addressing some of the limitations of the protocol and reducing the latency by combining the transport and security layers. QUIC also includes additional features such as connection migration and 0-RTT handshakes that are not available in TCP. Since the standardization of the QUIC protocol, a number of studies have been conducted to evaluate its security properties and to detect potential attacks either on the protocol itself or over it [20], [7]. One of the attacks that has been identified in the RFC [14] is the use of the QUIC handshake to perform Denial of Service (DoS) through traffic amplification. This attack is based on the fact that to reduce the latency of the connection establishment, the TLS certificate of the server can be sent at the beginning of the handshake without verifying the client's IP address. This can result in a large volume of data being sent to a targeted device. The QUIC specification includes a mechanism called Retry to prevent this attack by sending a token to the client, token that must be included in an additional response to the server before the handshake can continue. However, this requires an additional round trip, going against the low-latency principle of QUIC causing some servers to disable this feature. The QUIC RFC [14] describes a comparable vulnerability that emerges when utilizing QUIC's connection migration functionality. In a QUIC connection, each endpoint is identified by a set of connection IDs (CID) allowing to change the IP address of the client or the server without interrupting the connection. By changing the IP address of the client during a download, an attacker can redirect the data sent by the server to a target. To prevent this attack, the QUIC specification includes a path validation mechanism that requires the client to send a random value to the server to prove that it is the legitimate endpoint. This verification also takes an additional round trip and can be used as an attack vector on the server itself by sending a large number of path validation requests to the server [27]. This could be a reason for servers not to support path validation and consequently be potential amplifiers. In this paper we study whether this is the case.

978-3-903176-74-4 ©2025 IFIP

Besides the path validation mechanism, the QUIC specification states that a QUIC endpoint must limit the amount of data sent to an unvalidated address to three times the amount of data received from that address. This limit applies both to the handshake and in the case of a migration. However, at the time of writing, discussions are ongoing to make this limit five times the amount of data received [3] to better fit the current practices of top domains studied [19].

Lately, the deployment of QUIC servers increased from two Million QUIC responsive IP addresses in 2021 to more than 12 M in 2024 [33], [34]. In addition, the connection migration, is supported by a one Million distinct IP addresses [6]. Furthermore, new optimizations are being developed to improve the performance of QUIC implementations, increasing the throughput for legitimate use but also for potential attackers [29], [31], [15]. It is becoming more important to avoid vulnerabilities as public QUIC servers could be exploited as potential reflectors in DoS attacks causing a significant impact on the Internet.

This paper presents a study of the current state of QUIC servers and how they implement the mechanisms to prevent DoS attacks. Section II presents the background needed to understand the rest of the paper such as the QUIC protocol and a more detailed presentation of the vulnerabilities that are studied. Section III presents other studies related to QUIC measurements and DoS attacks. Section IV presents the methodology used to conduct the study as well as our experiment setup. Section V and Section VI present the results of the studies on the handshake and migration amplification DoS attacks respectively. Section VII concludes the paper.

II. Background

This section provides a concise overview of the QUIC protocol. We present mechanisms essential for understanding the remainder of the paper.

QUIC connection: QUIC is a connection-oriented protocol. Unlike other protocols that rely solely on IP addresses and port numbers, QUIC explicitly identifies connections by including a CID in each packet, allowing for connection migration. The first CID used to identify each side of the connection is included in the header of the Initial packet sent by the endpoint. After the handshake, endpoints can use *New Connection IDs* and *Retire Connection IDs* frames to add or remove CIDs to the connection. When an endpoint uses a different interface, it is recommended to change the CIDs associated to both endpoints. This practice enhances security by making it difficult for potential eavesdroppers to link packets sent over different interfaces to the same connection [11].

QUIC handshake: The QUIC handshake includes a TLS 1.3 handshake, the client sends a first packet containing a Client Hello message to the server. If no previous connection has been established, the destination CID is randomly generated. The message also contains a source CID that the server can use to send packets back to the client as well as the parameters supported by the client. If the server accepts the connection, it replies with a Server Hello message containing a TLS 1.3 Server Hello message and the transport parameters. Alternatively, to avoid sending the full Server Hello on an unverified path, and contribute to an amplification attack, the server can send a Retry packet with a token that has to be echoed by the client in order to continue the handshake. The header of the first packet sent by the server uses the CID provided by the client as destination CID and indicates the source CID that the client should use to send packets back to the server. An example of QUIC handshakes with and without Retry is shown in Figure 1.

QUIC connection migration: QUIC connection migration relies on the CIDs to allow endpoints to change their IP address or port without interrupting the connection. Migration can be either active or passive. An active migration is initiated by the client by using the new interface to send a *Path Challenge* frame containing a random nonce to the server. The server replies with a *Path Response* frame containing the same nonce to validate the new path. This requires that both endpoints have at least one unused CID available to send packets to the other endpoint.

Passive migration occurs when an endpoint's IP address changes due to a middlebox or network failure. In this case, the CID remains the same. Here, the endpoint that observes the change can send a *Path Challenge* frame with a random nonce. An endpoint that receives a *Path Challenge* must reply with a *Path Response* frame containing the nonce to validate the new path. Data can still be sent over the new path before validation as long as the volume of data sent (in bytes) respects the amplification limit currently set to 3.

Amplification attacks: The goal of an amplification attack is to overwhelm a target by using a reflector that sends a large volume of data to the target while sending the smallest possible amount of data from the attacker.

Because the TLS handshake is directly included in the QUIC handshake, the server can send a large TLS certificate in response to a single packet from the client. An amplification attack exploits this by sending the initial QUIC packet to the server while spoofing the IP address of the target, causing the server to redirect the response to the target. The QUIC client initial packet has a minimum size of 1,200 bytes. The QUIC specification states that an endpoint should not send more than three times the number of bytes received from an unverified path, 3,600 bytes in this case. However, TLS certificates used in QUIC handshakes are often larger than this limit and servers tend to send full certificates regardless of the size of the initial packet sent by the client [21]. Overall, there are several mechanisms that can limit the data sent by the server depending on the Server Hello total size. If the whole Server Hello fits in 3.600 bytes, it can be sent directly. The QUIC amplification limit should prevent servers from sending more but if it's ignored, then the only limit left is the congestion window of the servers.

When the server supports connection migration, an attacker can simulate an IP address change by sending spoofed packets on an already established connection. The attacker has to first

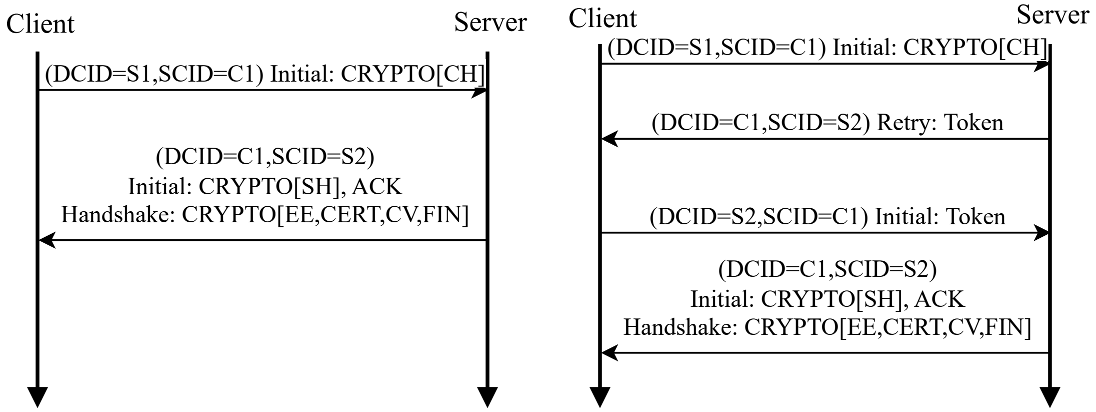

Client
Server
(DCID=S1,SCID=C1) Initial: CRYPTO[CH]
(DCID=C1,SCID=S2)
Initial: CRYPTO[SH], ACK
Handshake: CRYPTO[EE,CERT,CV,FIN]
Client
Server
(DCID=S1,SCID=C1) Initial: CRYPTO[CH]
(DCID=C1,SCID=S2) Retry: Token
(DCID=S2,SCID=C1) Initial: Token
(DCID=C1,SCID=S2)
Initial: CRYPTO[SH], ACK
Handshake: CRYPTO[EE,CERT,CV,FIN]

Fig. 1: Example of a QUIC handshake without Retry on the left and with Retry on the right. The Retry mechanism is the first DoS prevention mechanism studied in this paper. It lengthens the handshake by one RTT.

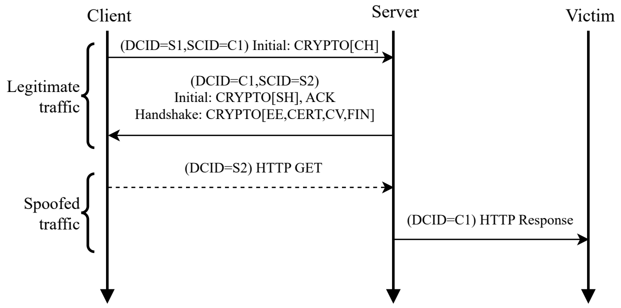


Fig. 2: Examples of the QUIC migration attack. Dashed lines represent spoofed packets.

complete the handshake with the server then sends the actual request with the victim's IP address as source to redirect the server's response to the target. Servers should limit the amount of data sent on the new path until the path is validated through the path validation mechanism. An example of the migration attack is shown in Figure 2.

The QUIC specification states that when a migration is performed, the congestion control mechanism should be reset to avoid congestion on the new path but there is no study on whether this is enforced by servers or not. Because the responses from servers can become quite large when requesting a video or a large file, the congestion control might be the only limiting factor for the amount of data sent by the server.

Both attacks have defense mechanisms that can be deployed to prevent them. Source IP address spoofing can be prevented by using SAV to verify that the source IP address is legitimate. The QUIC handshake includes a retry mechanism that can be used to ensure that the client is responsive before sending large volumes of data. The connection migration uses path validation to make sure that a new path is valid before sending data on it. In this paper we focus on the deployment of the last two techniques.

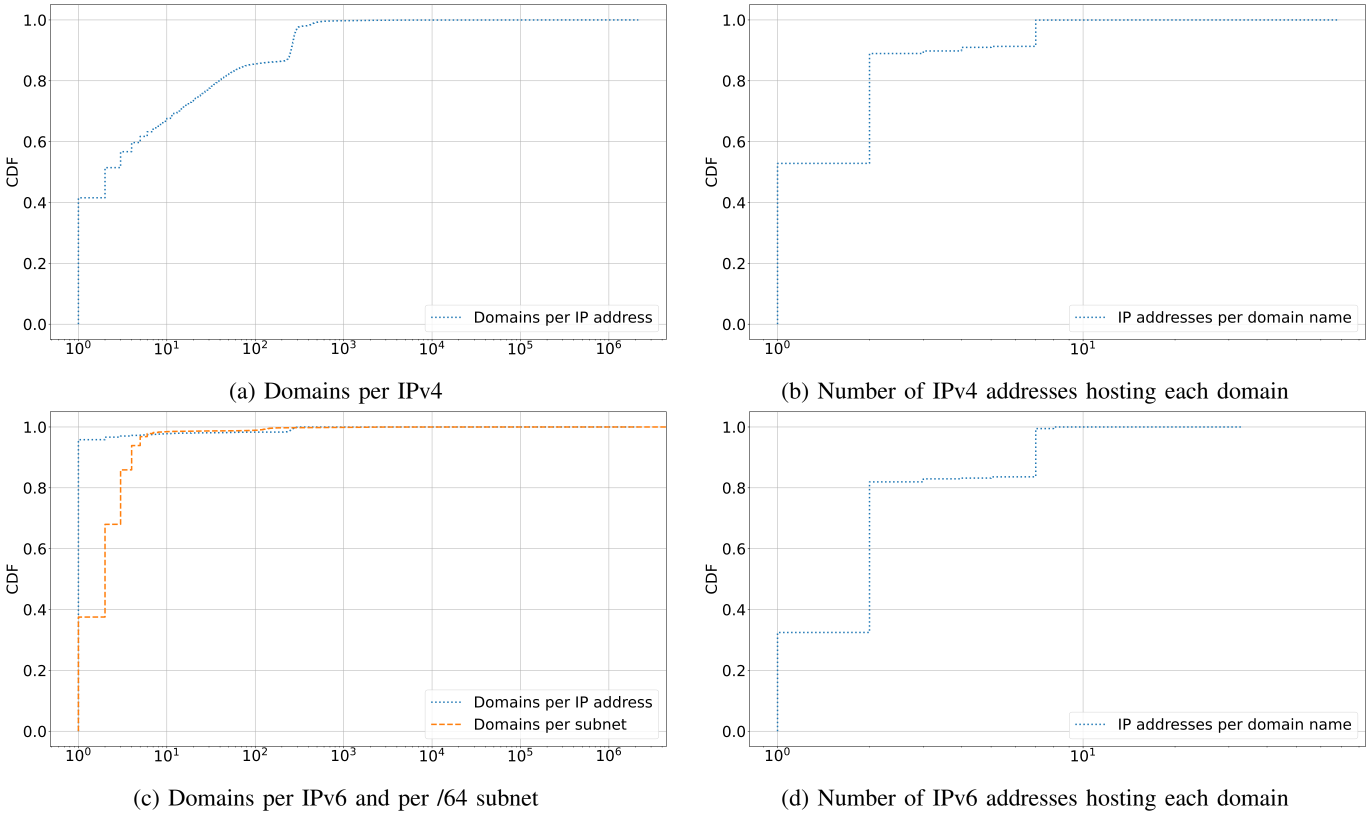
III. Related work

Several active scan studies were conducted to verify various properties of QUIC servers: Explicit congestion notification by Sander et al. [25], connection migration by Buchet et al. [6] and TLS certificate properties by Nawrocki et al. [21]. While these studies show a wide deployment of QUIC servers, none of them evaluate the amplification factor of QUIC servers in the context of DoS attacks using connection migration. Microsoft is performing periodic reachability test for the top 5000 hostnames and reports the number of servers breaking the amplification limit [19]. Their data shows that a majority of tested servers are breaking the limit but it's limited to 5000 domains and only tests the handshake not the migration. A study from Nawrocki et al. [20] collected backscatter data from a network telescope to identify DoS attacks on QUIC servers. The study focuses on attacks targeting QUIC servers directly such as flood attacks and considered amplification attacks to be unlikely at that time. Zirngibl et al. [34] proposed a method to identify QUIC libraries used by servers allowing more precise categorization of QUIC traffic. Unfortunately, the security of a server cannot be entirely determined by the library identified as there might be custom configurations, modifications or different versions of the library in use.

IV. Methodology

This section presents the methodology used in our study, from data acquisition to the measurement setup. We discuss ethical considerations in Appendix A.

Data acquisition: The study is focused on QUIC HTTP/3 servers that are publicly accessible. Previous studies showed that a Server Name Indication (SNI) has to be provided in order to be able to establish a connection with most QUIC servers [6], [34]. Hence, the starting point of the data acquisition is obtaining a large set of domain names. We combined the Tranco top list [17] with domain name lists from top level (com, net, org) domains from the Centralized Zone Data Service (CZDS) to reach a total of nearly 200M domain names.



(a) Domains per IPv4

(b) Number of IPv4 addresses hosting each domain

(c) Domains per IPv6 and per /64 subnet

(d) Number of IPv6 addresses hosting each domain

Fig. 3: Number of domain names hosted on each IP address and number of IP addresses hosting each domain name.

We resolved the IP addresses of these domains for both IPv4 and IPv6 using MassDNS [4] with an Unbound local resolver. We then used zmap [9] to test responsiveness of the IP addresses to QUIC connections. The zmap scans rely on the QUIC version negotiation mechanism, using reserved versions that trigger a version negotiation packet from the server. The presence of the negotiation indicates that QUIC is supported by the host on the scanned port.

Once we identified QUIC responsive IP addresses and their corresponding domain names, we performed stateful scans using a QUIC client, based on Cloudflare quiche [8], to simulate the different attacks. The client can be used to send spoofed handshake packets or perform full handshakes before sending a spoofed request. The source code of the tool will be provided on GitHub upon acceptance of the paper and when we have ensured that it cannot be used for malicious purposes.

In order to reduce the number of targets to scan and avoid overloading servers, we only considered each IP address once, despite an IP address often hosting multiple SNIs. When a high number of domain names are hosted on the same IP address, it is likely that the QUIC server and configuration will be the same. Accumulating results for such IP addresses would introduce a bias and would not be representative of the general behavior of all QUIC servers. We verified the exact spread of the domain names over the IP addresses space and the number of IP addresses hosting each domain name. This study gives us an idea of the spread of filtering we apply when we select a single domain per IP. It also shows that the bias that may be introduced in the data by multiple domains hosted on different IPs. The results are shown in Figure 3. The left side of the figure shows the number of domain names hosted on each QUIC responsive IP address tested. For IPv4, 60% of the IP addresses host more than one domain name with a significant number of IP addresses (18%) hosting more than 100 domain names. There are also around 10 IP addresses that are hosting more than 1 M domain names with a maximum at 2.1 M on a single IP address. For IPv6, there are way more IP addresses hosting a single domain name, with 98% of the IP addresses hosting only one domain name. There are still a few IP addresses hosting 2 M domain names. With the IPv4 address space being scarce, it is common for a single IP address to identify the webserver(s) in an entire LAN whereas in IPv6 network operators can assign a unique address to each web domain hosted in a /64 prefix. Grouping the IPv6 addresses by /64 prefixes, we observe a repartition of the number of domain names hosted on each prefix similar to the one observed for IPv4. We posit that most of the servers have the same behavior for all domains hosted on the same IP address and subnet. The consistency of the results at the granularity of /64 prefixes is verified in Section V.

The right side of Figure 3 shows the number of IP addresses hosting each domain name tested. A majority, 88% (for IPv4) and 81% (for IPv6) of the domain names are hosted on one or two IP addresses. The proportion of domains hosted on two different IP addresses is higher for IPv6 at 45% compared to 33% for IPv4. The maximum number of IP addresses that are hosting a particular domain name is around 800 for IPv4 and just over 300 for IPv6 so considering a domain name multiple times would not have a significant impact on the results.

Based on these observations, we selected unique pairs of IP addresses and domain names to ensure comprehensive and representative measurements while avoiding redundant scans on the same server. For IPv6, we also present results at the /64 prefix level to correct biases arising from subnets hosting a large number of domain names.

Measurement setup: The experiment setup consists of a pair of VMs, hosted in the same cloud environment, in our infrastructure in North America. We configured one VM as the attacker and one as the target of the attack. The attacker is sending the spoofed packets to the remote servers while the target, in the same cloud, is used to receive the data returned by the servers. In order to limit the risk of the attacks failing because of spoofing protection, the attacker and the target VMs are run in the same subnet. The setup is shown in Figure 4. The scans were performed over a the span of a week in March 2024. The collection of data is done by capturing the UDP packets sent and received on both VMs using tcpdump.

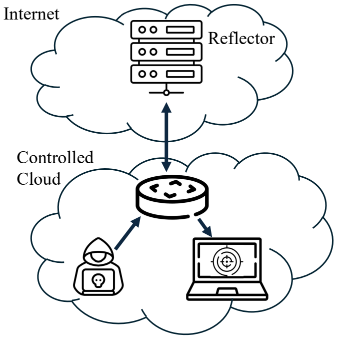
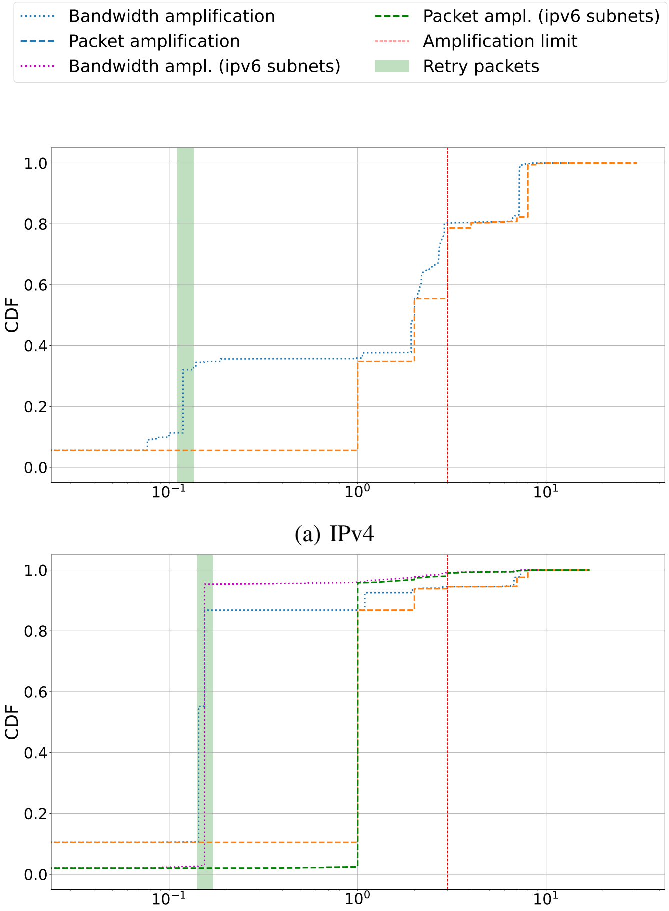


Fig. 4: Measurement setup for the study. The attacker and target VMs are hosted in the same cloud environment.

V. Handshake amplification DoS

From the zmap scans and the DNS resolution, we identified around 450 K IPv4 targets and 2.9 M IPv6 targets within 470 K /64 subnets. Each target is a unique pair of an IP address responsive to QUIC connections and a domain name. The high number of v6 targets comes from the fact that way more domain names are hosted on a single IPv6 address than on an IPv4 address as shown in section IV. This can be due to the fact that it is easier to get a large number of IPv6 addresses as IPv4 addresses are becoming scarce.

We sent spoofed QUIC initial packets towards all the targets from the attacker VM and measured the traffic that was sent back towards the victim VM. The CDFs for the bandwidth and number of packets amplification factors observed are shown in Figure 5. In IPv6 we present results at the IP and /64 prefix level. When servers are grouped by /64 we plot the mean amplification experienced for the servers in the prefix. This curve corrects biases arising from highly responsive prefixes [26]. Among the 470 k subnets, the standard deviation of the results is null for 460 k of them, showing that the behavior of the servers is consistent within the same subnet.



(a) IPv4

(b) IPv6

Fig. 5: Amplification factor of the handshake attack. 20% of the IPv4 servers are above the limit, most IPv6 servers/subnets used Retry.

There are no packets received from 5% of the IPv4 targets and 2% of the IPv6 subnets. This could be due to changes in the servers configurations between the zmap scan and the stateful scan as there has been several hours between the two. The targets using the Retry mechanism demonstrate a packet amplification factor of one. We observe that, for IPv4, around 30% of the responsive servers are just sending a Retry packet. For IPv6 this number goes up to 83% of servers and 93% of the subnets using the Retry mechanism.

The overall bandwidth amplification factor of the handshake attack remains quite low. The average for IPv4 is at 2.4 and for IPv6, the overall average for all targets is only at 0.6 while state-of-the-art amplification attacks are typically way higher [12]. It is interesting to note that despite previous studies exposing servers breaking the amplification limit [21], the QUIC specification is still not enforced for all servers at the time of writing. We observe that around 20% of IPv4 servers still have an amplification factor over 3 with some going even over 10. It is also worth noting that almost all the servers over the current limit would also be over the proposed limit of 5 [3]. For IPv6, most of the servers not using Retry are located in a few subnets corresponding to 17% of the total number of IP addresses. But the amplification factors remain for the most part under the limit with only 9% of the servers over 3. Only 2% of the subnets present at least one server over the limit but less than 1% of them have a mean amplification factor over 3.

QUIC Providers

TABLE I: Mean packet/bandwidth amplification with the standard deviation provided in parenthesis (std) for the 5 organizations with the most targets for the handshake attack. For IPv6, the first row for the targets is the total number of targets and the second row is the number of subnets, in gray.

IPv4

Organization	**Targets** (IP with domain)	**Mean Packet Amplification (Std)**	**Mean Bandwidth Amplification (Std)**
Cloudflare	116,180 (26.1%)	5.6 (2.5)	5.3 (2.9)
Hostinger	108,882 (24.5%)	1.1 (0.5)	0.3 (0.4)
AWS	35,824 (8.0%)	1.6 (2.3)	1.3 (2.5)
Hengda	17,590 (4.0%)	2.9 (0.3)	2.1 (0.4)
Google	14,422 (3.2%)	2.8 (1.1)	2.5 (0.9)

IPv6

Organization	**Targets** (IP with domain)	**Mean Packet Amplification**	**Mean Bandwidth Amplification**
Hostinger	2,245,408 (76.6%) 440,312 (93.1%)	1.0 (0.0)	0.1 (0.1)
AWS	549,554 (18.7%) 23,837 (5.0%)	1.7 (2.4)	1.4 (2.5)
Cloudflare	105,315 (3.6%) 104 (0.0%)	5.96 (2.5)	5.5 (2.8)
PrivateSyst.	7,345 (0.2%) 84 (0.0%)	2.96 (0.2)	2.7 (0.2)
CRI-AS	4,884 (0.2%) 5 (0.0%)	2.4 (0.9)	2.0 (1.1)

In order to identify the organizations behind the servers, we used the data from the RouteViews project [24]. We retrieve a BGP full feed from a collector in the same region as the attacker VM and used it to map each IP address to an AS number. We then used the data from the Caida AS to organization mapping [30] to go from the AS number to the name of the organization responsible for the IP address. The top organizations in terms of the number of targets are shown in Table I.

For IPv4, the top provider, Cloudflare has a mean amplification factor of 5.3 which is slightly above the limit of the QUIC specification but still quite low. It is also the provider with the highest absolute standard deviation indicating that some of their servers are way above the limit. The second provider, Hostinger, has a way lower amplification factor of 0.3 indicating that most of their servers use the Retry mechanism. The standard deviation is also quite low indicating that the behavior of the servers is consistent. These two providers represent the top and bottom 25%, respectively, of the servers in terms of amplification factor, in our results. The other providers are way smaller in terms of the number of targets and tend to be somewhere in between the two extremes with AWS having the highest standard deviation relative to the mean which indicates that their servers are more spread out in terms of amplification factor.

For IPv6, the results are heavily dominated by Hostinger, representing 76.6% of the targets and 93.1% of the subnets. The behavior of Hostinger for IPv6 is consistent with the one observed for IPv4 with a mean amplification factor of 0.1 indicative of the use of the Retry mechanism. Cloudflare remains the top provider with the highest average bandwidth amplification factor at 5.5, still above the limit of the QUIC specification.

VI. Migration amplification DoS

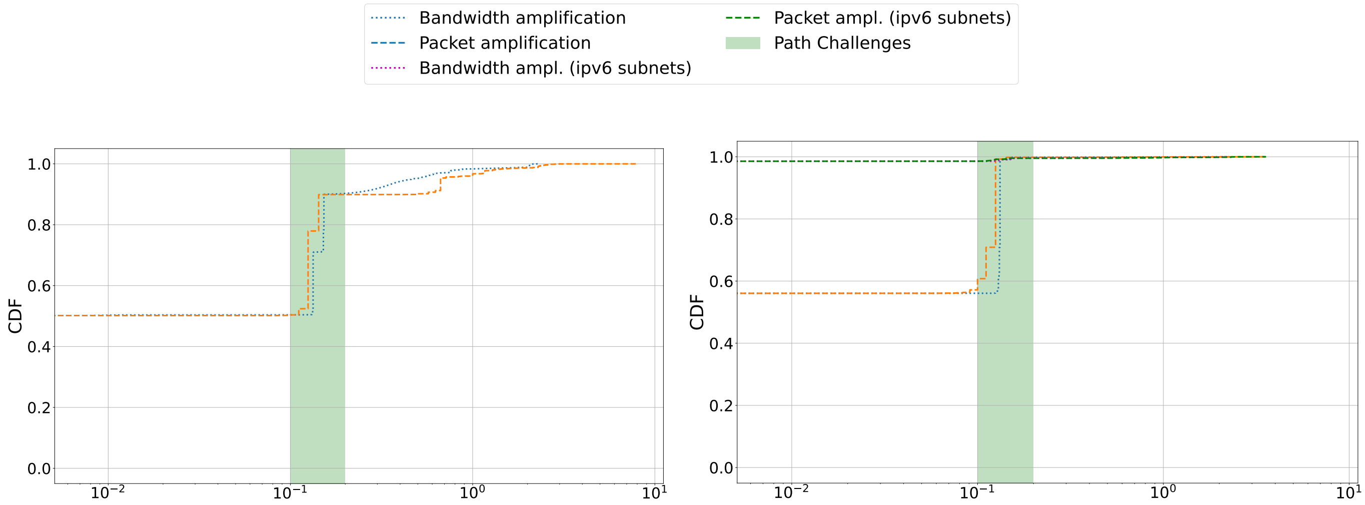
Here, we performed the migration by using a different IP address for the client as soon as the QUIC handshake was completed without using the active migration mechanism. This form of migration, while less secure than active migration because the two flows can be linked by an eavesdropper, has the advantage of not requiring additional CIDs exchange which the server might not perform. It is also easier to handle for load balancers and middleboxes that might only rely on the CIDs to match packets to connections.

The spoofed traffic sent consisted only of a simple HTTP GET request for the root of the domain while the handshake was done in a regular way using the IP address of the "attacker" VM.

The CDFs for the bandwidth and number of packets amplification factors observed are shown in Figure 6.

The first thing to note is that while the handshake attack only required any kind of response from the server to create a reflection, this attack requires both a successful handshake and a response to the migration attempt. In total, we didn't receive any response on the victim VM from around half of the contacted IPv4 servers. This lack of response was mostly due to the handshake not being completed. For the 450 K IPv4, the handshake was completed with 230 K of the servers. For IPv6, the number of servers responding was concentrated within a very small number of subnets with only 4% of subnets responding. When a response was sent, for most of the servers, it consisted of only a single packet with a path challenge on the new path. It appears that although the specification allows sending up to 3 times the amount of data received, most servers choose to avoid sending anything on unvalidated paths. For the few servers that sent more than just a path challenge, the amplification factor remains extremely small with 99% of the servers not even breaking the x1 mark making the attack counterproductive. The overhead of performing a complete handshake before actually performing the migration makes it hard to create an efficient attack.

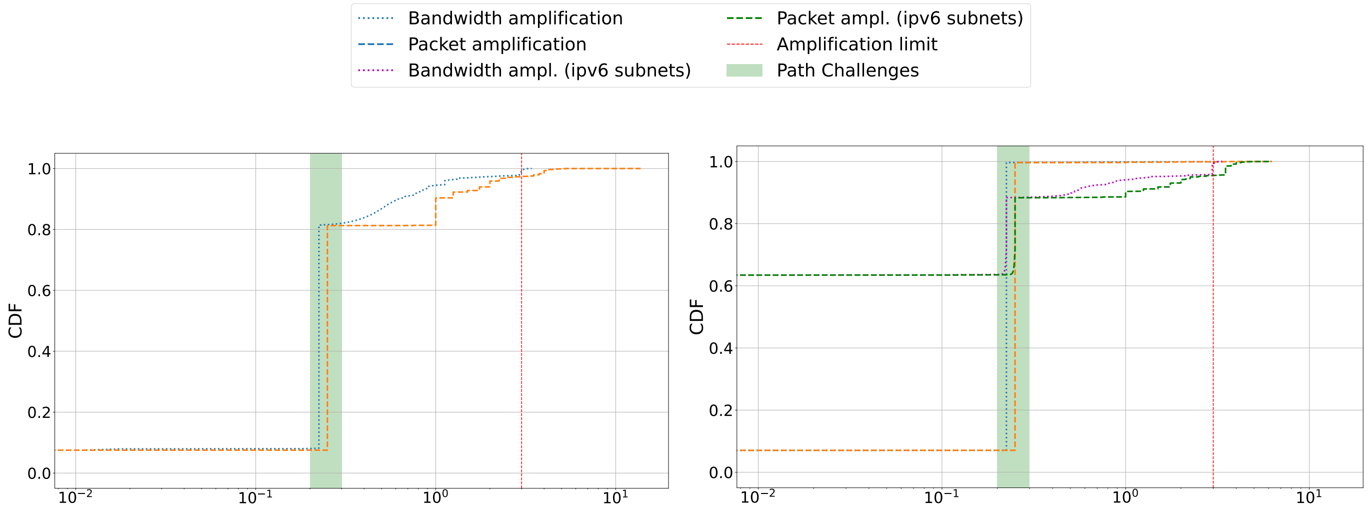
The QUIC RFC only explicitly limits the amount of data sent on unvalidated paths so we also looked at the



(a) IPv4

(b) IPv6

Fig. 6: Amplification factor of the migration attack. Around half of IPv4 targets are not responding to the attack. Most of the servers that responded sent only a path challenge and no additional data, leading to low amplification factors.



(a) IPv4

(b) IPv6

Fig. 7: Amplification factor of the migration attack only considering the spoofed traffic. Most of the IPv4 for which a packet is sent respond with a path challenge. Around 20% of them send more data. Only 40% of the IPv6 subnets send a response. Among them, more than half only send a path challenge. The small wall matching the amplification limit might indicate an explicit restriction on the amount of data sent.

amplification factor of the spoofed traffic only. The results for targets for which spoofed traffic was sent are shown in Figure 7. To focus on how servers respond to passive migration, we filter out servers for which the handshake was not completed as there was no spoofed traffic sent to them. This highlights that most of the servers are only sending a path challenge and no additional data. For IPv6, we can see that only 37% (4k/11k) of the subnets are sending traffic at the victim VM but these subnets represent 93% (538k/578k) of the total number of IP addresses. A few servers are still sending more data than the amplification limit, showing that the limit is not enforced by all servers. Even servers breaking the limit have very limited amplification factors ($<$ 10) and are probably not worth using as reflectors for DoS attacks. The observed reflected traffic was too limited to assess whether congestion control misconfiguration would affect the attack, as traffic volumes from the servers never reached the typical initial values for the congestion window [13].

QUIC Providers

Similarly to the handshake, we identified the organizations behind the servers used with the migration attack. Counting only servers for which the handshake was completed, the top organizations are shown in Table II. The values are only reported for the spoofed traffic.

For IPv4, Hostinger is by far the top provider and has a mean amplification factor very low at 0.2. This factor is aligned with other providers and shows that most of the servers are only sending a path challenge and no additional data. Google is the only provider in the top 5 that has a mean amplification factor above 1 but it remains quite low at 1.5. It is also the provider with the highest standard deviation while the other providers have a standard deviation close to 0 indicating that the behavior of the servers is consistent.

For IPv6, Hostinger is the first provider in terms of the number of targets with more than 90% of the total targets but not in the number of subnets with only 14%. Their servers behavior is the same as for IPv4 with a mean amplification factor of 0.2. AWS is the second provider in terms of number of targets but the first in terms of subnets with 61% of the total. Unfortunately, there was no packet reflected from their servers when attempting a passive migration. The other providers have a mean amplification factor of 0.2 or 0.3 showing once again that most of the servers are only sending a path challenge with eventually a retranmission. Some of the servers are sending an actual HTTP response but they are very few and the amplification factor never reaches levels that would make a DoS attack efficient. All IPv6 providers have a standard deviation very close to 0 showing that the behavior of the servers is consistent for each organisation.

TABLE II: Mean packet/bandwidth amplification with the standard deviation (std) provided in parenthesis for the 5 organizations with the most targets for the migration attack. For IPv6, the first row for the targets is the total number of targets and the second row is the number of subnets, in gray. Amplification factors are only reported for the spoofed traffic.

IPv4

Organization	**Targets**	**Mean Packet Amplification (Std)**	**Mean Bandwidth Amplification (Std)**
Hostinger	100,189 (41.82%)	0.2 (0.1)	0.2 (0.1)
Hengda	17,197 (7.2%)	1.0 (0.3)	0.5 (0.3)
Google	11,883 (5.0%)	2.6 (1.0)	1.5 (1.1)
A2Hosting	9,331 (3.9%)	0.2 (0.0)	0.2 (0.0)
OVH	6,514 (2.7%)	0.4 (0.4)	0.3 (0.5)

IPv6

Organization	**Targets**	**Mean Packet Amplification**	**Mean Bandwidth Amplification**
Hostinger	533,072 (92.1%) 1,602 (14.1%)	0.2 (0.0)	0.2 (0.0)
AWS	35,033 (6.1%) 6,957 (61.1%)	null (null)	null (null)
PrivateSyst.	3,180 (0.5%) 81 (0.7%)	0.3 (0.1)	0.2 (0.1)
CRI-AS	1,405 (0.2%) 3 (0.0%)	0.2 (0.0)	0.2 (0.0)
GuzelHost.	817 (0.1%) 9 (0.1%)	0.2 (0.0)	0.2 (0.0)

VII. Conclusion

This study presents a view of the current state of QUIC servers and how they deploy protections against possible amplification DoS attacks. We tested two different types of attacks, one relying on the handshake and the other using the connection migration mechanism, looking at the spread of the Retry and path validation mechanisms as well as the compliance with the amplification limit of the QUIC protocol standard. We collected data from more than 450 K IPv4 and 2.9 M IPv6 targets and performed the attacks on controlled targets to measure the responses from the servers. The results show that while the overall amplification factor of the attacks remains quite low, 20% of IPv4 and 9% of IPv6 servers do not follow the general guidelines regarding the limit of data sent to unverified IP addresses. The Retry handshake mechanism is used by around 30% of IPv4 servers and 75% of IPv6 servers, while the path validation mechanism used to prevent migration attack is enforced by most of the servers that support connection migration. With the increase in the deployment of QUIC servers and connection migration, we believe that continuously monitoring the danger of potential attacks and the deployment of existing protections is important.

VIII. Acknowledgments

This research was partly supported by the European Union and Wallonia in the framework of the program "Wallonia 2021-2027"

References

[1] Manish Arora, Shawn Bohrer, Omer Yoachimik, Cody Doucette, Alex Forster, and Nick Wood. How Cloudflare auto-mitigated world record 3.8 Tbps DDoS attack. https://blog.cloudflare.com/how-cloudflare-auto-mitigated-world-record-3-8-tbps-ddos-attack/.

[2] Fred Baker and Pekka Savola. Ingress Filtering for Multihomed Networks. RFC 3704, March 2004.

[3] Nick Banks. Proposal: Increase quic amplification limit to 5x. IETF Mail Archive, 2024. https://mailarchive.ietf.org/arch/msg/quic/Qc1C-TP3tsvQ1i_-uEDSIU3iH0c/.

[4] Birk Blechschmidt and Quirin Scheitle. Massdns. https://github.com/blechschmidt/massdns.

[5] Kevin Bock, Abdulrahman Alaraj, Yair Fax, Kyle Hurley, Eric Wustrow, and Dave Levin. Weaponizing middleboxes for TCP reflected amplification. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 3345–3361. USENIX Association, August 2021.

[6] Aurélien Buchet and Cristel Pelsser. An Analysis of QUIC Connection Migration in the Wild, 2024. https://arxiv.org/abs/2410.06066.

[7] Efstratios Chatzoglou, Vasileios Kouliaridis, Georgios Karopoulos, and Georgios Kambourakis. Revisiting quic attacks: A comprehensive review on quic security and a hands-on study. *International Journal of Information Security*, 22(2):347–365, 2023.

[8] Cloudflare. Cloudflare Quiche. https://github.com/cloudflare/quiche.

[9] Zakir Durumeric, Eric Wustrow, and J Alex Halderman. ZMap: fast internet-wide scanning and its security applications. In *22nd USENIX Security Symposium (USENIX Security 13)*, pages 605–620, 2013.

[10] Wesley Eddy. Transmission Control Protocol (TCP). RFC 9293, August 2022.

[11] Yashodhar Govil, Liang Wang, and Jennifer Rexford. {MIMIQ}: Masking {IPs} with migration in {QUIC}. In *10th USENIX Workshop on Free and Open Communications on the Internet (FOCI 20)*, 2020.

[12] Salih Ismail, Hani Ragab Hassen, Mike Just, and Hind Zantout. A review of amplification-based distributed denial of service attacks and their mitigation. *Computers & Security*, 109:102380, 2021.

[13] Jana Iyengar and Ian Swett. QUIC Loss Detection and Congestion Control. RFC 9002, May 2021.

[14] Jana Iyengar and Martin Thomson. QUIC: A UDP-Based Multiplexed and Secure Transport. RFC 9000, May 2021.

[15] Benedikt Jaeger, Johannes Zirngibl, Marcel Kempf, Kevin Ploch, and Georg Carle. Quic on the highway: evaluating performance on high-rate links. In *2023 IFIP Networking Conference (IFIP Networking)*, pages 1–9. IEEE, 2023.

[16] Erin Kenneally and David Dittrich. The menlo report: Ethical principles guiding information and communication technology research. *Available at SSRN 2445102*, 2012.

[17] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczynski, and Wouter Joosen. Tranco: A research-oriented top sites ranking hardened against manipulation. In *Proceedings 2019 Network and Distributed System Security Symposium*, San Diego, CA, 2019. Internet Society.

[18] Qasim Lone. Sav: Why is source address validation still a problem? RIPE Labs, 2023. https://labs.ripe.net/author/qasim-lone/sav-why-is-source-address-validation-still-a-problem/.

[19] Microsoft. QUICReach. https://microsoft.github.io/quicreach/.

[20] Marcin Nawrocki, Raphael Hiesgen, Thomas C. Schmidt, and Matthias Wählisch. Quicsand: quantifying quic reconnaissance scans and dos flooding events. In *Proceedings of the 21st ACM Internet Measurement Conference*, IMC '21, page 283–291, New York, NY, USA, 2021. Association for Computing Machinery.

[21] Marcin Nawrocki, Pouyan Fotouhi Tehrani, Raphael Hiesgen, Jonas Mücke, Thomas C Schmidt, and Matthias Wählisch. On the interplay between tls certificates and quic performance. In *Proceedings of the 18th International Conference on emerging Networking EXperiments and Technologies*, pages 204–213, 2022.

[22] Yevheniya Nosyk, Maciej Korczyński, and Andrzej Duda. Routing loops a mega amplifiers for dns-based ddos attacks. In Oliver Hohlfeld, Giovane Moura, and Cristel Pelsser, editors, *Passive and Active Measurement*, pages 629–644, Cham, 2022. Springer International Publishing.

[23] Yevheniya Nosyk, Maciej Korczyński, Qasim Lone, Marcin Skwarek, Baptiste Jonglez, and Andrzej Duda. The closed resolver project: Measuring the deployment of inbound source address validation. *IEEE/ACM Transactions on Networking*, 31(6):2589–2603, 2023.

[24] University of Oregon. Route views project. http://www.routeviews.org/routeviews/.

[25] Constantin Sander, Ike Kunze, Leo Blöcher, Mike Kosek, and Klaus Wehrle. Ecn with quic: Challenges in the wild. In *Proceedings of the 2023 ACM on Internet Measurement Conference*, pages 540–553, 2023.

[26] Patrick Sattler, Johannes Zirngibl, Mattijs Jonker, Oliver Gasser, Georg Carle, and Ralph Holz. Packed to the brim: Investigating the impact of highly responsive prefixes on internet-wide measurement campaigns. *Proceedings of the ACM on Networking*, 1(CoNEXT3):1–21, 2023.

[27] Marteen Seemann. Exploiting quic's path validation, 2023. https://seemann.io/posts/2023-12-18---exploiting-quics-path-validation/.

[28] Daniel Senie and Paul Ferguson. Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. RFC 2827, May 2000.

[29] Nikita Tyunyayev, Maxime Piraux, Olivier Bonaventure, and Tom Barbette. A high-speed quic implementation. In *Proceedings of the 3rd International CoNEXT Student Workshop*, pages 20–22, 2022.

[30] The CAIDA UCSD. As to organization mapping dataset, 01/02/2025-28/02/2025. https://www.caida.org/catalog/datasets/as-organizations.

[31] Xiangrui Yang, Lars Eggert, Jörg Ott, Steve Uhlig, Zhigang Sun, and Gianni Antichi. Making quic quicker with nic offload. In *Proceedings of the Workshop on the Evolution, Performance, and Interoperability of QUIC*, pages 21–27, 2020.

[32] Omer Yoachimik and Jorge Pacheco. DDoS threat report for 2024 Q2. https://blog.cloudflare.com/ddos-threat-report-for-2024-q2/.

[33] Johannes Zirngibl, Philippe Buschmann, Patrick Sattler, Benedikt Jaeger, Juliane Aulbach, and Georg Carle. It's over 9000: analyzing early QUIC deployments with the standardization on the horizon. In *Proceedings of the 21st ACM Internet Measurement Conference*, page 261–275, Virtual Event, Nov 2021. ACM.

[34] Johannes Zirngibl, Florian Gebauer, Patrick Sattler, Markus Sosnowski, and Georg Carle. QUIC Hunter: Finding QUIC Deployments and Identifying Server Libraries Across the Internet. In *International Conference on Passive and Active Network Measurement*, pages 273–290. Springer, 2024.

Appendix

Ethics

The study was conducted following the best practices for ethical Internet measurements [16]. The attacker VM is rate-limited to avoid overloading our providers. The number of packets send to the scanned servers is very low, only 1 for the handshake and less than 10 for the migration. The traffic sent to each server consisted of a simple handshake with a small HTTP request in the case of the migration which should not cause any perturbation on the servers. The target of the attack was directly under our control and we made sure that our scans did not cause any perturbation on the network hosting the VMs. There are web servers hosted on the two VMs with a description of the study and a contact email allowing for opt-out requests. We maintained a list of domains and prefixes that requested to be excluded from the study and avoided scanning them. We are planning to contact the owners of servers that presented a high amplification factor to inform them of the potential risk of their servers being used in an attack. We will help providers willing to verify that their servers can no longer be used for amplification attacks. Because we are not sure all concerned servers will take into account our request, we prefer not to disclose the full list of servers identified with the highest amplification factors at the moment to limit the risk of them being used in an attack. No personal data was collected during the study.