MASQUE QUIC-Aware CONNECT-UDP Encryption Design Team Update

draft-ietf-masque-quic-proxy

IETF 118 – Prague – 2023-11

Tommy Pauly, David Schinazi, Ben Schwartz,
Antoine Fressancourt, Eric Rosenberg, Mirja Kühlewind

Scope of the design team work

- Work on the threat model for “forwarded” proxying mode and compare to standard “tunneled” mode for UDP proxying
- Work on proposals for adding encryption, and analyze how these impact the security and privacy properties of the protocol

Executive Summary

1. We analyzed passive and active attacks for both standard UDP proxying and Forwarded Mode
2. We propose an extensible re-encryption model for forwarding (“packet transforms”)
3. We propose an initial re-encryption mode, “scramble”, that protects clients from the byte-matching passive attacks

Threat Analysis

Threat model

The attacker’s goal is to violate CONNECT-UDP’s privacy properties, i.e. to learn which targets are being accessed by each client.

- This is equivalent to learning a mapping between a Connection ID seen on the proxy-to-target path and a Connection ID on the client-to-proxy path

Threat model analysis:
Attacker description

Review of academic work around deanonymization attacks on Tor and other Anonymous Communication Network (ACN systems) considers two types of attackers:

- A global ***passive*** attacker, able to eavesdrop any link
- A global ***active*** attacker, able to inject, drop, copy or delay packets

⇒ Some work shows that eavesdropping on some links close to the source is as powerful as a global passive attacker

⇒ Active attacks can be performed just by dropping packets

Threat model analysis:
Attacks performed

In the last few years, deanonymization attacks on Tor and other ACN systems are based on traffic analysis:

- Obvious attacks based on packet metadata observation have been prevented by changes in protocol and header formats
  ⇒ *Packets crossing a proxy need to be protected from such naïve attacks*

- Traffic correlation attacks use the timing, inter-packet arrival, or packet size differences to correlate packet flows together
  ⇒ *Need to allow the use of padding to act on packet size*
  ⇒ *Explicit protection by introducing delay or actively interleaving packets are out of design team scope*

Let’s go through how passive and active attacks apply to UDP proxying

As we go through these: which attacks have we missed?

Passive attacks

Goal of the attacker is to correlate traffic across both sides of the proxy

	Tunneled / RFC 9298	Forwarded
Recognizing matching bytes in packets	✅ Not vulnerable	❌ Vulnerable without encryption!
Recognizing packets based on timing or size (exact or close)	❌ Vulnerable	❌ Vulnerable
Mappings between CIDs on client-to-proxy and proxy-to-target (attacker analysis is different)	Many to one	One to one

Proposal: “Scramble” encryption

- Unauthenticated, length-preserving encryption using AES-128.
- Scrambled packets follow QUIC invariants; all bits scrambled except the Connection ID and the “Header Form bit”.
- Can be implemented in a single forward pass.
- Construction is similar to QUICv1 Header Protection.
- Supports all current QUIC versions, but not all *possible* versions.
- Doesn’t currently explicitly add padding or chaff (could be added as an extension)

https://github.com/ietf-wg-masque/draft-ietf-masque-quic-proxy/pull/87

Passive attacks, updated

Goal of the attacker is to correlate traffic across both sides of the proxy

	Tunneled / RFC 9298	Forwarded + Scramble
Recognizing matching bytes in packets	✅ Not vulnerable	✅ Not vulnerable
Recognizing packets based on timing or size (exact or close)	❌ Vulnerable	❌ Vulnerable
Mappings between CIDs on client-to-proxy and proxy-to-target (correlation is different)	Many to one	One to one

Active attacks

Goal of the attacker is to correlate traffic across both sides of the proxy

	Tunneled / RFC 9298	Forwarded + Scramble
Inject packets from client to proxy with a known CID, to recognize on the other side	✅ Not vulnerable	❌ Requires authentication
Inject one or more replayed packets from client to proxy, to recognize a burst on the other side	✅ Not vulnerable	❌ Requires anti-replay
Intercept packets from client to proxy, and corrupt some, to see which are dropped from the other side	❌ Vulnerable	✅ Not vulnerable
Inject a burst of packets from target to proxy with a known CID, to recognize on the other side	❌ Vulnerable	❌ Vulnerable

We also discussed congestion-based attacks, which will behave differently with forwarded mode (that doesn't add extra congestion control) and tunneled mode. Forwarded mode may have a slight advantage.

Should we handle active attacks?

Both Tunneled and Forwarded mode are vulnerable to active attackers. The set of attacks is different, but there are attacks possible on both sides of the proxy.

It would be possible to have a forwarding mode with truncated authentication tags and counters, but such a mode resembles a reimplementation of QUIC. We are inclined to recommend just the Scramble approach, since we're vulnerable to active attackers anyway.

These attacks will determine which deployment scenarios are appropriate for forwarding mode (e.g. access network or dual/multiple proxy setup), with detailed discussion in security considerations.

Impacts of correlation attacks

When identifying targets reveals which end servers (websites, etc) a user is accessing, the correlation attack could directly expose user activity

When the target is another proxy hop, especially one that many or all other clients of this proxy use, the correlation attack doesn't directly expose any sensitive information.

- If the attacker *is* the next hop proxy, or colluding with the next hop proxy, they can break the overall privacy via correlation. However, the next hop proxy is in a position to be a powerful active attacker who can correlate with or without forwarding mode.