[**Version Downgrade Prevention**](https://datatracker.ietf.org/doc/html/rfc9368#name-version-downgrade-preventio) **from RFC 9368 (may 2023)**

**Downgrade Attack Context**  
A version downgrade attack occurs when an attacker manipulates negotiation so that QUIC endpoints end up using a version other than the one they would have selected under normal conditions. The document outlines mechanisms designed to prevent such attacks.

**Ignoring Certain Version Negotiation Packets**

Clients MUST ignore Version Negotiation packets that include the Original Version.

If a client starts a new connection based on a Version Negotiation packet, it MUST ignore any subsequent Version Negotiation packets received for that new attempt.

**Validating Version Information**

Both client and server MUST parse each other’s Version Information during the handshake. If parsing fails (e.g., incorrect length), the connection MUST be closed.

A Chosen Version or Available Version set to zero also causes a parsing failure, leading to a connection close.

If the server sees that the Chosen Version is not in its list of Available Versions, it MUST close the connection.

QUIC versions that support version negotiation MUST define an error for version negotiation failures. In QUIC v1, this is the VERSION\_NEGOTIATION\_ERROR.

**Server Validation**

The server determines the version in use (e.g., by checking the Version field in the Long Header) and then verifies that the client’s Chosen Version matches.

If they differ, the server closes the connection.

Specifically in QUIC v1, if the first Long Header packet has Version 0x00000001, but the client’s Chosen Version is not 0x00000001, the server MUST close the connection.

**Handling Missing Version Information**

Servers MAY complete the handshake even if Version Information is missing.

Clients reacting to a Version Negotiation packet MUST NOT complete the handshake if Version Information is missing; otherwise, they MAY do so.

**Client Validation**

If the server’s Chosen Version was not among the client’s Available Versions, the client MUST close the connection.

If the client reacted to a Version Negotiation packet but the server’s Version Information is missing, it MUST close the connection.

Clients MUST validate the server’s Available Versions to confirm they would have chosen the same version if those versions had been advertised. If not, the client MUST close with a version negotiation error.

**Example Scenarios**

The client starts with version 12, the server only fully deploys 13 and 14, and sends a Version Negotiation packet listing 10, 13, and 14. The client then picks 14, which matches what it would have chosen if it had known about 13 and 14 from the start, so the handshake proceeds with version 14.

An attacker forges a Version Negotiation packet listing 10 and 13. The client picks 10. However, the server’s Available Versions include 10, 13, and 14, meaning the client would actually have preferred 14. This discrepancy results in a version negotiation error.

**Final Version in Use**

Once the process completes, the version in use is the one indicated as the “Chosen Version” by the server.

The client MUST check that the server’s Chosen Version matches the Negotiated Version. If not, the client closes the connection.

Overall, these rules ensure that endpoints detect and ignore malicious attempts to downgrade QUIC to an unintended version.