CONNECT-UDP - with Binding support

The client can open UDP port(s) on the proxy to receive traffic from multiple targets.
Enabled by setting the target to * and providing the connect-udp-bind header in
the Request Headers.

HEADERS

:method = CONNECT
:protocol = connect-udp
:scheme = https
:path = /masque/udp/*/*/
:authority = proxy.org
capsule-protocol = ?1
connect-udp-bind = ?1
What’s New? Target Compression

- Define two new capsules
  - COMPRESSION_ASSIGN: Allocate Context IDs to targets
  - COMPRESSION_CLOSE: Release Context IDs

CAPSULE COMPRESSION_ASSIGN {
  Context ID (i),
  IP Version (8),
  IP Address (32..128),
  UDP Port (16),
}

CAPSULE COMPRESSION_CLOSE {
  Context ID (i),
}
COMPRESSION_ASSIGN Capsule

CAPSULE COMPRESSION_ASSIGN {
    Context ID (i),
    IP Version (8),
    IP Address (32..128),
    UDP Port (16),
}

- Client or proxy can request compression of a target via this capsule
- Other party confirms allocation by echoing back the capsule
- When IP Version is set to 0, an Uncompressed Context is requested
Uncompressed Contexts

CAPSULE COMPRESSION ASSIGN {
    Context ID (i) = 2,
    IP Version = 0
}

When IP Version is set to zero in COMPRESSION_ASSIGN, this context to allow connections from/to any target IP/Port
Target IP and Port MUST be provided on each frame

Only the client can allocate these.
Compressed Contexts

CAPSULE COMPRESSION ASSIGN {
  Context ID (i) = 2,
  IP Version (8),
  IP Address (32..128),
  UDP Port (16),
}

DATAGRAM QUIC Frame {
  Type (i) = 0x30..0x31,
  [Length (i)],
  Quarter Stream ID (i),
  Context ID (i) = 2,
  UDP Payload (...),
}

Once the COMPRESSION capsule is exchanged, all frames exchanged between the proxy and client can omit the Target IP:Port, and use Context ID instead while it is valid.
Releasing a Context

CAPSULE COMPRESSION_CLOSE {
    Context ID (i) = 2,
}

Can be sent by either client or proxy, to clean up or close connections. Resource management for both parties

This is also sent to reject a COMPRESSION_ASSIGN request. If for example, the proxy is unable to deal with more connections.
Achieves 2 goals at once

1) COMPRESSION: Save bytes from sending IP:Port per datagram using context IDs
2) IP RESTRICTION: Remove context IDs to reject traffic from a given target or all uncompressed targets. Create them to poke holes in the “firewall”
What’s new? The Proxy Public address header

The proxy is now required to return IP:Port tuples to the client in Response Headers. Multiple ones can be provided

HEADERS

:status = 200
capsule-protocol = ?1
connect-udp-bind = ?1
proxy-public-address = 192.188.0.2:8211, [2001:db8::1234]:54321
Allow more than one address per family?

We decided to allow more than one. ICE/WebRTC are designed to function with multiple candidate IP addresses.
Identifying the proxy local address used

We allow the proxy to pick its own proxy local address for relaying payloads.

Is knowledge of control of which specific proxy local address the client picked useful to the client?

Should the client be able to select the address it wants per packet/context?
MASQUE
CONNECT-UDP
Bind
draft-ietf-masque-connect-udp-listen

IETF 120 – Vancouver – 2024-07-23

David Schinazi – dschinazi.ietf@gmail.com
Abhi Singh - abhisingh.ietf@gmail.com