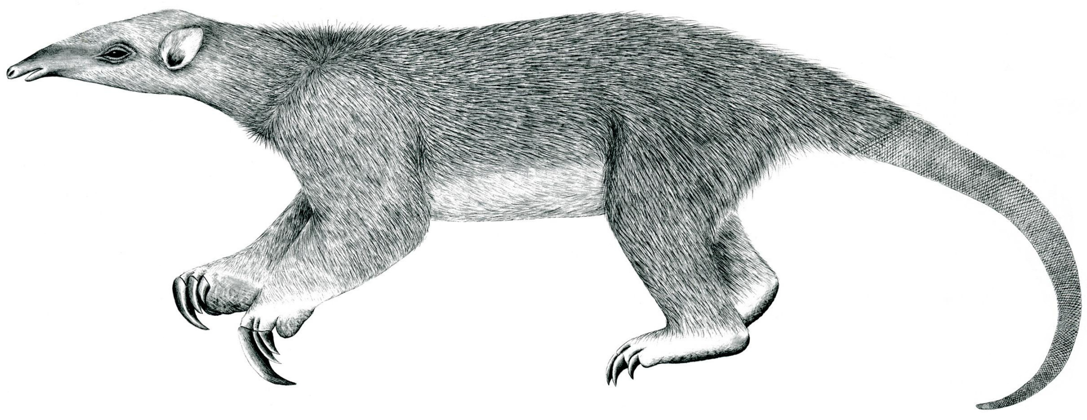
Tranalyzer2

quicDecode

QUIC (IETF)



Tranalyzer Development Team

Contents

1 quicDecode

1.1 Description

The quicDecode plugin analyzes QUIC (IETF) traffic.

1.2 Dependencies

If `QUIC_DECODE_TLS=1`, then **libssl** is required.

		QUIC_DECODE_TLS=1
Ubuntu:	`sudo apt-get install`	`libssl-dev`
Arch:	`sudo pacman -S`	`openssl`
openSUSE:	`sudo zypper install`	`libopenssl-devel`
Red Hat/Fedora[1]**:**	`sudo dnf install`	`openssl-devel`
macOS[2]**:**	`brew install`	`openssl@1.1`

1.3 Configuration Flags

The following flags can be used to control the output of the plugin:

Name	**Default**	**Description**
`QUIC_SPKT_TYPE_STR`	1	Format of packet type in packet mode 0: number, 1: string
`QUIC_DECODE_TLS`	1	0: do not decrypt QUIC Initial packets 1: decrypt TLS 1.3 handshake in QUIC Initial packets Use with the `sslDecode` plugin to extract the SNI and JA3 fingerprint
`QUIC_DEBUG`	0	0: do not print any debug messages 1: print debug messages

1.4 Flow File Output

The quicDecode plugin outputs the following columns:

Column	**Type**	**Description**	**Flags**
`quicStat`	H8	Status	
`quicVersion`	H32	Version	
`quicFlags`	H8	Flags	
`quicPktTypes`	H8	Packet Types	
`quicDCID`	SC	Destination Connection ID	
`quicSCID`	SC	Source Connection ID	

[1]If the `dnf` command could not be found, try with `yum` instead
[2]Brew is a packet manager for macOS that can be found here: https://brew.sh

Column	Type	Description	Flags
`quicDCID`	SC	Original Destination Connection ID (Retry)	

1.4.1 quicStat

The `quicStat` column is to be interpreted as follows:

quicStat	Description
`0x01`	Flow is QUIC
`0x02`	Handshake (Packet Type is 2)
`0x04`	Version negotiation (version is 0)
`0x08`	Version changed
`0x10`	Destination Connection ID changed
`0x20`	Source Connection ID changed
`0x40`	Original Destination Connection ID changed
`0x80`	Packet was snapped (t2buf failed)

1.4.2 quicVersion

The `quicVersion` column is to be interpreted as follows[3]:

quicVersion	Description
`0x00000000[0001-ffff]`	Standardized versions of QUIC
`0x454747[00-ff]`	NetApp quant
`0x50435130`	Private Octopus Picoquic internal test version
`0x5130303[1-9]`	Google QUIC 01–09 (Q001–Q009)
`0x5130313[0-9]`	Google QUIC 10–19 (Q010–Q019)
`0x5130323[0-9]`	Google QUIC 20–29 (Q020–Q029)
`0x5130333[0-9]`	Google QUIC 30–39 (Q030–Q039)
`0x5130343[0-9]`	Google QUIC 40–49 (Q040–Q049)
`0x51474f[00-ff]`	quic-go (QGO[0–255])
`0x91c170[00-ff]`	quicly (qicly0[0–255])
`0xabcd000[0-f]`	Microsoft WinQuic
`0xf10000[00-ff]`	IETF QUIC-LB
`0xf123f0c[0-f]`	Mozilla MozQuic
`0xfaceb00[0-f]`	Facebook mvfst
`0xff[000000-ffffff]`	IETF QUIC draft-xx[4]
`0xf0f0f0f[0-f]`	ETH Zürich Measurability experiments
`0xf0f0f1f[0-f]`	Telecom Italia Measurability experiments

[3]For a more exhaustive list, refer to `https://github.com/quicwg/base-drafts/wiki/QUIC-Versions`
[4]The latest draft is `draft-ietf-quic-transport-34` with version `0xff000022`

1.4.3 quicFlags

The `quicFlags` column is to be interpreted as follows:

quicFlags	Description
`0x03`	Packet Number Length (Long Header)
`0x0c`	Reserved (Long Header)
`0x20`	Spin Bit (Short Header)
`0x30`	Packet Type (Long Header)
`0x40`	Fixed Bit
`0x80`	Long Header

1.4.4 quicPktTypes

The `quicPktTypes` column is to be interpreted as follows:

quicPktTypes	Description
2^0 `(=0x01)`	Initial
2^1 `(=0x02)`	0-RTT
2^2 `(=0x04)`	Handshake
2^3 `(=0x08)`	Retry

1.5 Packet File Output

In packet mode (`-s` option), the quicDecode plugin outputs the following columns:

Column	Type	Description	Flags
`quicFlags`	H8	Flags	
`quicPktType`	S/U8	Packet Type	
`quicVersion`	SC	Version	
`quicDCID`	SC	Destination Connection ID	
`quicSCID`	SC	Source Connection ID	
`quicODCID`	SC	Original Destination Connection ID	
`quicPktNum`	U32	Packet Number	

1.6 Plugin Report Output

The following information is reported:

- Number of QUIC packets
- Number of QUIC Initial packets
- Number of QUIC 0-RTT packets
- Number of QUIC Handshake packets
- Number of QUIC Retry packets

1.7 Known Bugs and Limitations

- The quicDecode plugin assumes every UDP packet on port 443 or 4433 from after 2015[5] is QUIC…

1.8 References

- draft-ietf-quic-applicability: Applicability of the QUIC Transport Protocol
- draft-ietf-quic-bit-grease: Greasing the QUIC Bit
- draft-ietf-quic-datagram: An Unreliable Datagram Extension to QUIC
- draft-ietf-quic-http: Hypertext Transfer Protocol Version 3 (HTTP/3)
- draft-ietf-quic-invariants: Version-Independent Properties of QUIC
- draft-ietf-quic-load-balancers: QUIC-LB: Generating Routable QUIC Connection IDs
- draft-ietf-quic-manageability: Manageability of the QUIC Transport Protocol
- draft-ietf-quic-qpack: QPACK: Header Compression for HTTP/3
- draft-ietf-quic-recovery: QUIC Loss Detection and Congestion Control
- draft-ietf-quic-tls: Using TLS to Secure QUIC
- draft-ietf-quic-transport: QUIC: A UDP-Based Multiplexed and Secure Transport
- draft-ietf-quic-version-negotiation: Compatible Version Negotiation for QUIC

[5] QUIC was submitted for standardization to the IETF in 2015