QUIC FIXES FOR NETWORK SECURITY MONITORING

David McGrew, PhD
mcgrew@cisco.com
linkedin.com/david-mcgrew-949b80
hnull.org
January 12, 2023

QUESTIONS

- Who uses QUIC and why?
- How and why does QUIC inhibit security monitoring?
- Can we extract metadata and fingerprints from QUIC?
- What should security teams do with QUIC?

https://github.com/cisco/mercury/ open source and Network Protocol Fingerprinting

QUIC

OVERVIEW

- Google proprietary from 2012 (gQUIC)
- IETF draft (2015) and RFC 9000 (2021)
- Replaces TCP, runs over UDP 443
- Incorporates/replaces TLS

STATUS

- Supported by Google, Apple, Microsoft, Cloudflare, ...
- Used on over half of sessions in Chrome/Google ecosystem
- UDP/443 blocked by many enterprises

PLUSES AND MINUSES

BENEFITS

- Lower latency than TLS/TCP
- Multiplexing without head of line blocking
- ‘Anti-ossification’

DEMERITS

- Challenges network visibility through encryption, fragmentation, and randomization
- Challenges traditional network and operating system access control

QUIC AND TLS HAVE A COMPLICATED RELATIONSHIP

```
+-------------+-------------+ +-------------+
|     TLS     |     TLS     | |    QUIC     |
|  Handshake  |    Alerts   | | Applications|
|             |             | |  (h3, etc.) |
+-------------+-------------+-+-------------+
|                                           |
|               QUIC Transport              |
|   (streams, reliability, congestion, etc.) |
|                                           |
+-------------------------------------------+
|                                           |
|          QUIC Packet Protection           |
|                                           |
+-------------------------------------------+
```

ANTI-OSSIFICATION

- Problem: operating systems and ‘middleboxes’ inhibit protocol evolution
  - TCP Fast Open - slow to get in Linux kernel, blocked by firewalls
  - SCTP non-adoption
- Solution:
  - Encrypt all messages
  - ‘Greasing’ and randomization of messages
- RFC 8558: *The IAB urges protocol designers to design for confidential operation by default. We strongly encourage developers to include encryption in their implementations and to make them encrypted by default.*
- RFC 9170: *[Grease] reserves values for extensions that have no semantic value attached.*

QUIC ANTI-VISIBILITY FEATURES

- *Every* packet is encrypted (except version negotiation)
- Padding to maximum packet length
- TLS client hellos needlessly fragmented*
- TLS extensions and QUIC transport parameters randomly shuffled*

* *Not part of standard; observed with Chrome and other clients*

INITIAL PACKET ENCRYPTION

- *Every* packet is encrypted
- Initial packets encrypted with fixed, well-known key
- Keys are *not* registered with IANA

INITIAL SALTS

```
quic_initial_salt = {
    {4207849473, salt_enum::D22},      // faceb001
    {4207849474, salt_enum::D23_D28}, // faceb002
    {4207849486, salt_enum::D23_D28}, // faceb00e
    {4207849488, salt_enum::D23_D28}, // faceb010
    {4207849489, salt_enum::D23_D28}, // faceb011
    {4207849490, salt_enum::D23_D28}, // faceb012
    {4207849491, salt_enum::D23_D28}, // faceb013
    {4278190102, salt_enum::D22},      // draft-22
    {4278190103, salt_enum::D23_D28}, // draft-23
    {4278190104, salt_enum::D23_D28}, // draft-24
    {4278190105, salt_enum::D23_D28}, // draft-25
    {4278190106, salt_enum::D23_D28}, // draft-26
    {4278190107, salt_enum::D23_D28}, // draft-27
    {4278190108, salt_enum::D23_D28}, // draft-28
    {4278190109, salt_enum::D29_D32}, // draft-29
    {4278190110, salt_enum::D29_D32}, // draft-30
    {4278190111, salt_enum::D29_D32}, // draft-31
    {4278190112, salt_enum::D29_D32}, // draft-32
    {4278190113, salt_enum::D33_V1},  // draft-33
    {4278190114, salt_enum::D33_V1},  // draft-34
    {1,          salt_enum::D33_V1},  // version-1
};
```

```
std::array<salt, 4> salts{
  salt{{0x7f,0xbc,0xdb,0x0e,0x7c,0x66,0xbb,0xe9, 0x19,0x3a,
        0x96,0xcd,0x21,0x51,0x9e,0xbd,0x7a,0x02,0x64,0x4a}, "d22"},
  salt{{0xc3,0xee,0xf7,0x12,0xc7,0x2e,0xbb,0x5a,0x11,0xa7,
        0xd2,0x43,0x2b,0xb4,0x63,0x65,0xbe,0xf9,0xf5,0x02}, "d23_d28"},
  salt{{0xaf,0xbf,0xec,0x28,0x99,0x93,0xd2,0x4c,0x9e,0x97,
        0x86,0xf1,0x9c,0x61,0x11,0xe0,0x43,0x90,0xa8,0x99}, "d29_d32"},
  salt{{0x38,0x76,0x2c,0xf7,0xf5,0x59,0x34,0xb3,0x4d,0x17,
        0x9a,0xe6,0xa4,0xc8,0x0c,0xad,0xcc,0xbb,0x7f,0x0a}, "d33_v1"}
};
```

QUIC PACKET FORMATS

```
Initial Packet {
  Header Form (1) = 1,
  Fixed Bit (1) = 1,
  Long Packet Type (2) = 0,
  Reserved Bits (2),
  Packet Number Length (2),
  Version (32),
  Destination Connection ID Length (8),
  Destination Connection ID (0..160),
  Source Connection ID Length (8),
  Source Connection ID (0..160),
  Token Length (i),
  Token (..),
  Length (i),
  Packet Number (8..32),
  Packet Payload (8..),  # Encrypted sequence of frames
}
```

```
Version Negotiation Packet {
  Header Form (1) = 1,
  Unused (7),
  Version (32) = 0,
  Destination Connection ID Length (8),
  Destination Connection ID (0..2040),
  Source Connection ID Length (8),
  Source Connection ID (0..2040),
  Supported Version (32) ...,
}
```

PLAINTEXT EXAMPLE: PADDING AND FRAGMENTATION

```
01000000000006406240480606010201003300260024001d00207d133aa81de3d8f392de8ac8baf883be42e1a9b30935b1c9a4ae2b223c950935000000160014
0000117777772e696e7374616772616d2e636f010640e8146e74656c204d6163204f5320582031305f31355f0640fc404d37030245c080ff73db0c00000001fa
baca7a0000000108024064200480010000070480600000e47b74db52b8321602547e0604806000000f000902406744690005000302683 3002b00030203040101
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000001061601b706000d0100014503032a4dd793e7b2450617404b044f9119fa01ee34b088bf76860e990000061301130213
0301000116001b0003020002000a00080006001d00170018001000050003026833000d0014001204030804040105030805050108060d090ab86c22d6ed919599
01000000000000000000000000000000000000000000000000000000010101000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000640aa3e6d002d00020101003900840404 80f0000080004752040000
0001050480600000010480007530712 92a4368726f6d652f39392e302e343834342e373420490000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000
```

QUIC METADATA

Field	Location	Notes
Server Name	TLS Client Hello	Optional
Application Layer Protocol	TLS Client Hello	
Google User Agent	TLS Client Hello	Optional
Error Messages	Connection Close	

EXAMPLE METADATA: QUIC INITIAL PACKET


```
{
  "tls": {
    "client": {
      "version": "0303",
      "random": "2a4dd793e7b2450ab86c22d6ed919599b7044f9119fa01ee34b088bf76860e99",
      "session_id": "",
      "cipher_suites": "130113021303",
      "compression_methods": "00",
      "server_name": "www.instagram.com",
      "quic_transport_parameters": "00390084040480f0000080004752040000000010504806000000104800075307129 2a4368726f6d6...",
      "google_user_agent": "Chrome/99.0.4844.74 Intel Mac OS X 10_15_7",
      "application_layer_protocol_negotiation": [
        "h3"
      ]
    }
  },
  "quic": {
    "connection_info": "11001100",
    "version": "00000001",
    "dcid": "a54e31098418c193",
    "scid": "",
    "token": "",
    "data": "5e585ea4694c66b20967dede634bdd9052b1be26cd461e47486d1179cb0ec08379e20ade289dfbadf2948a5f6d7157c1098613...",
    "salt_string": "d33_v1"
}
```

EXAMPLE METADATA: QUIC INITIAL PACKET


```
{
  "tls": {
    "client": {
      "version": "0303",
      "random": "0ae2cac4b506c0d8ee963f9f0b7cf728cac5d6a6d579fb28eb6c127026f0c01d",
      "cipher_suites": "130113031302",
      "compression_methods": "00",
      "server_name": "i.ytimg.com",
      "quic_transport_parameters": "0039005380ff73db0c000000019a3a0a9a00000001c0000000ff02de1a0243e80c006ab200040481...",
      "application_layer_protocol_negotiation": [
        "h3"
      ]
    }
  },
  "quic": {
    "connection_info": "11000011",
    "version": "00000001",
    "salt_string": "d33_v1",
  },
  "src_ip": "10.26.160.131",
  "dst_ip": "142.251.16.119",
  "protocol": 17,
  "src_port": 57211,
  "dst_port": 443,
}
```

EXAMPLE METADATA: CONNECTION CLOSE MESSAGE


```
{
  "quic": {
    "connection_info": "11001101",
    "version": "00000001",
    "connection_close": {
      "error_code": 0,
      "frame_type": 0,
      "reason_phrase": "25:No recent network activity after 4016800us. Timeout:4snum_undecryptable_packets: 0 {}"
    },
    "salt_string": "d33_v1",
  },
  "src_ip": "64.100.12.5",
  "dst_ip": "172.253.122.94",
  "protocol": 17,
  "src_port": 22751,
  "dst_port": 443,
}
```

APPLICATIONS USING QUIC

Chromium	Web Browsers
Firefox	Web Browsers
Apple	Networking
Syncthing	File Synchronization
IPFS	File Synchronization
Deimos	Command and Control Framework
Merlin	Command and Control Framework
Psiphon	Censorship Circumvention Tool
Malware	

APPLICATION LAYER PROTOCOL NEGOTIATION (ALPN)

Observed ALPN Values	Notes
["h3"]	HTTP3
["h3-29"]	HTTP3 Variant
["h3-alias-01"]	HTTP3 Variant
["h3","h3-29"]	HTTP3 Variants
["h2","http/1.1","quic"]	HTTP Variants
["http/1.1","h2","h3"]	HTTP Variants
["h3-fb-05"]	Proprietary HTTP3
["bep/1.0"]	Block Exchange Protocol (Syncthing)
["doq"]	DNS over QUIC
["smb"]	Server Message Block 2 (Microsoft)

TLS ALPN: list of protocols advertised by client, in descending order of preference, named by IANA-registered, opaque, non-empty byte strings.

QUIC FINGERPRINTS

- Initial Message fingerprint requires key derivation, decryption, and defragmentation
- Fingerprint extends the TLS fingerprint definition
- TLS extensions and QUIC transport parameters are shuffled into random order by some clients
- Extensions are normalized by lexicographic sorting

Reference: https://github.com/cisco/mercury/blob/main/doc/npf.md

QUIC FINGERPRINTS

```
quic/
(ff00001d)
(0303)
(0a0a130113021303)
[ (0000)
  (000500050100000000)
  (000a000c000a0a0a001d001700180019)
  (000d0018001604030804040105030203080508050501080606010201)
  (00100008000605683332d3239)
  (0012)
  (001b0003020001)
  (002b0005040a0a0304)
  (002d00020101)
  (0033)
  (0a0a)
  (0a0a)
  (   (ffa5)
     [ (04)
       (05)
       (06)
     ]
  )
]
```

RANDOMIZED EXTENSIONS ARE NOT GOOD

- Complexity in security-critical code
- Untrusted code can use subliminal channel
  - Permutation of n elements can leak $\lg_2(n!)$ bits
  - Leak secret keys
  - User-tracking information
- Better idea: clients use lexicographic canonical ordering

https://hnull.org/2022/12/01/sorting-out-randomized-tls-fingerprints/

CONCLUSIONS

- QUIC improves on TCP, and its adoption will grow
- ‘Anti-ossification’ features compilcate visibility, but don’t prevent it
  - Application Layer Protocol, Server Name, User Agent, Cipher Suites, ...
  - Fingerprints
- Less appropriate for security-critical environments
  - Data Centers, Industrial Settings, IoT, Defense, Regulated Industries, ...

THANK YOU