Denial of Service vulnerability (infinite loop) while parsing malicious QUIC frame #969

Closed cve-reporting opened this issue on Jul 2, 2020 · 2 comments · Fixed by #970

Labels bug

cve-reporting commented on Jul 2, 2020

In picoquic QUIC server maliciously crafted QUIC frame triggers infinite loop while processing.

Incorrect logical conditions in picoquic_decode_frames() and picoquic_decode_stream_frame() leads to infinite loop after processing single packet in epoch==3.

Attack can be performed remotely without any user interaction and authentication.
Proposed CVSS 3.0 score: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H (7.5 - High)
(picoquic can be used in embedded environments where infinite loop in one module affects whole system, because there is not pre-emptive OS used)

Packet that triggers this issue (attached zipped): hang_001_decode_stream_frame.raw
To reproduce this issue in full server protocol session needs to be in state epoch==3.

Vulnerable loop is located in function picoquic_decode_frames() in picoquic/frames.c lines: 3630-3776.

Log of the loop with displayed parameters and variables:

```
picoquic/frames.c:3643 picoquic_decode_frames() : bytes: 0x20d2242 bytes_maxsize: 21
picoquic/frames.c:3630 picoquic_decode_frames() : bytes: 0x20d2242 bytes_maxsize: 21
picoquic/frames.c:902 picoquic_decode_stream_frame() : bytes 0x20d2242 bytes_max 255
picoquic/frames.c:918 picoquic_decode_stream_frame() : bytes 0x20d2242 | bytes_max 255 | stream_id 4611684923227504639 | offset 4611686018427387903 | data_length 2 | fin 1 | consumed 17
picoquic/frames.c:3643 picoquic_decode_frames() : bytes: 0x20d2242 bytes_maxsize: 21
picoquic/frames.c:3630 picoquic_decode_frames() : bytes: 0x20d2242 bytes_maxsize: 21
picoquic/frames.c:902 picoquic_decode_stream_frame() : bytes 0x20d2242 bytes_max 255
picoquic/frames.c:918 picoquic_decode_stream_frame() : bytes 0x20d2242 | bytes_max 255 | stream_id 4611684923227504639 | offset 4611686018427387903 | data_length 2 | fin 1 | consumed 17
...
```

Backtrace from gdb:

```
0x000000000041fb44 in picoquic_varint_decode (bytes=0x677251 "\377\377\377\377\377", bytes@entry=0x67724b '\377' <repeats 11 times>, max_bytes=max_bytes@entry=10,
n64=0x7fffffffd468)
at picoquic/picoquic/intformat.c:148
148 v += *bytes++;

(gdb) bt
#0 0x000000000041fb44 in picoquic_varint_decode (bytes=0x677251 "\377\377\377\377\377", bytes@entry=0x67724b '\377' <repeats 11 times>, max_bytes=max_bytes@entry=10,
n64=0x7fffffffd468) at picoquic/picoquic/intformat.c:148
#1 0x0000000000419b49 in picoquic_parse_stream_header (bytes=bytes@entry=0x677242 "\r\377\377\377\001", bytes_max=bytes_max@entry=19, stream_id=stream_id@entry=0x7fffffffd458,
offset=offset@entry=0x7fffffffd468, data_length=data_length@entry=0x7fffffffd460, fin=fin@entry=0x7fffffffd450, consumed=0x7fffffffd470)
at picoquic/picoquic/frames.c:645
#2 0x000000000041c3f1 in picoquic_decode_stream_frame (cnx=cnx@entry=0x694420, bytes=0x677242 "\r\377\377\377\001", bytes_max=bytes_max@entry=0x677255 "\377",
current_time=current_time@entry=0) at picoquic/picoquic/frames.c:909
#3 0x000000000041f731 in picoquic_decode_frames (cnx=cnx@entry=0x694420, path_x=0x694f30, bytes=, bytes@entry=0x677240 "\001", bytes_maxsize=bytes_maxsize@entry=21,
epoch=epoch@entry=3, addr_from=addr_from@entry=0x0, addr_to=0x0, current_time=0) at picoquic/picoquic/frames.c:3641
#4 0x0000000000415f67 in parse_frame_test (buffer=0x677240 "\001", byte_max=22) at picoquic/picoquicfirst/picoquicdemo.c:1100
#5 0x0000000000411fce in main (argc=, argv=) at picoquic/picoquicfirst/picoquicdemo.c:1261
```

```
(gdb) bt full
#0 0x0000000000041fb44 in picoquic_varint_decode (bytes=0x677251 "\377\377\377\377\377\377", bytes@entry=0x67724b '\377' <repeats 11 times>, max_bytes=max_bytes@entry=10,
n64=0x7fffffffd468) at picoquic/picoquic/intformat.c:148
i = 6
v = 70368744177663
length = 8
#1 0x0000000000419b49 in picoquic_parse_stream_header (bytes=bytes@entry=0x677242 "\r\377\377\377\001", bytes_max=bytes_max@entry=19, stream_id=stream_id@entry=0x7fffffffd458,
offset=offset@entry=0x7fffffffd468, data_length=data_length@entry=0x7fffffffd460, fin=fin@entry=0x7fffffffd450, consumed=0x7fffffffd470)
at picoquic/picoquic/frames.c:645
ret = 0
len =
off = 4
length = 0
l_stream = 8
l_len = 0
l_off = 0
byte_index = 9
FUNCTION = "picoquic_parse_stream_header"
#2 0x000000000041c3f1 in picoquic_decode_stream_frame (cnx=cnx@entry=0x694420, bytes=0x677242 "\r\377\377\377\001", bytes_max=bytes_max@entry=0x677255 "\377",
current_time=current_time@entry=0) at picoquic/picoquic/frames.c:909
stream_id = 4611684923227504639
data_length = 2
offset = 4611686018427387903
fin = 1
consumed = 17
#3 0x0000000000041f731 in picoquic_decode_frames (cnx=cnx@entry=0x694420, path_x=0x694f30, bytes=, bytes@entry=0x677240 "\001", bytes_maxsize=bytes_maxsize@entry=21,
epoch=epoch@entry=3, addr_from=addr_from@entry=0x0, addr_to=0x0, current_time=0) at picoquic/picoquic/frames.c:3641
first_byte =
bytes_max = 0x677255 "\377"
ack_needed =
pc = picoquic_packet_context_application
packet_data = {acked_path = 0x0, last_ack_delay = 0, last_time_stamp_received = 0, largest_sent_time = 0, delivered_prior = 0, delivered_time_prior = 0,
delivered_sent_prior = 0, rs_is_path_limited = 0}
FUNCTION = "picoquic_decode_frames"
#4 0x0000000000415f67 in parse_frame_test (buffer=0x677240 "\001", byte_max=22) at picoquic/picoquicfirst/picoquicdemo.c:1100
epoch = 3
ret = 0
simulated_time = 0
saddr = {sin_family = 0, sin_port = 0, sin_addr = {s_addr = 0}, sin_zero = "\000\000\000\000\000\000\000\000"}
qclient = 0x677670
FUNCTION = "parse_frame_test"
t_ret = 0
cnx = 0x694420
#5 0x0000000000411fce in main (argc=, argv=) at picoquic/picoquicfirst/picoquicdemo.c:1261
result = 0
```

Source code snippet to reproduce issue (rest of parameters are based on parse_frame_test() from picoquictest/skip_frame_test.c):

```
picoquic_decode_frames(cnx, cnx->path[0], buffer, byte_max, 3, NULL, NULL, simulated_time);
```

hang_001_decode_stream_frame.raw.zip

huitema added the bug label on Jul 2, 2020

huitema commented on Jul 2, 2020 Collaborator

Thanks for the report. I have a local repro and a fix, but I need to work on the tests a bit to make sure that there are no bugs in the same class lurking around.

huitema mentioned this issue on Jul 2, 2020

Reproduce and fix stream decode loop, add tests #970

Merged

huitema commented on Jul 2, 2020 Collaborator

@cve-reporting : The issue is being fixed in PR #970. Changes include:

1. Adding a manually compiled list of erroneous frames to skip_frame_test.c. This includes the frame in the attached file hang_001_decode_stream_frame.raw.zip.
2. Use that list for new sets of cases in `skip_frame_test()`, `parse_frame_test()`, `logger_test()` and `binlog_test()`. Verify that the additional tests in `parse_frame_test()` reproduce the loop issue.
3. Added a simple fuzz test in `parse_frame_test()`, just in case.
4. Verify that all tests pass, and that the loop issue in particular is fixed in frame.c.

huitema closed this as completed in #970 on Jul 2, 2020

Assignees

No one assigned

Labels

bug

Projects

None yet

Milestone

No milestone

Development

Successfully merging a pull request may close this issue.

Reproduce and fix stream decode loop, add tests
private-octopus/picoquic

2 participants