CVE-2020-24753: Memory corruption vulnerability in oocborrt

Summary:

A memory corruption vulnerability in Objective Open CBOR Run-time (oocborrt) before 2020-08-12 could allow an attacker to execute code via crafted Concise Binary Object Representation (CBOR) input to the cbor2json decoder. An uncaught error while decoding CBOR Major Type 3 text strings leads to the use of an attacker-controllable uninitialized stack value. This can be used to modify memory, causing a crash or heap corruption.

CVSS v3.1 Score: 6.5/10

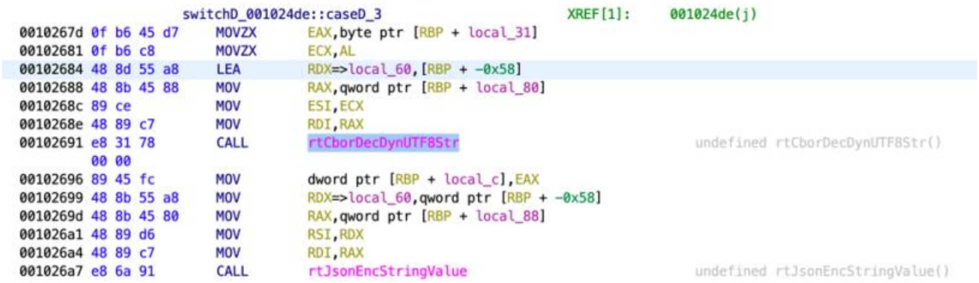
Vulnerability Details:

The cbor2json decoder utility in the Objective Open CBOR Run-time library up through commit de254ab (before 2020-08-12) is missing an error handling check after decoding a UTF8 string in a given CBOR data stream. Since this utility could be integrated into various data interchange systems which parse external inputs, a remote attacker may be able to execute arbitrary code through heap corruption.

cbor2json.c:

```
112:   case OSRTCBOR_UTF8STR: {
113:     OSUTF8CHAR* utf8str;
114:     ret = rtCborDecDynUTF8Str (pCborCtxt, ub, (char**)&utf8str;);
115:
116:     ret = rtJsonEncStringValue (pJsonCtxt, utf8str);
117:     rtxMemFreePtr (pCborCtxt, utf8str);
118:     if (0 != ret) return LOG_RTERR (pJsonCtxt, ret);
119:
120:      break;
121:   }
```

Line 115 of the above file is missing a check of the return value of rtCborDecDynUTF8Str(). If a malicious CBOR input causes rtCborDecDynUTF8Str() to return prematurely, the error is not caught and utf8str is erroneously passed to rtJsonEncStringValue(). utf8str is attacker-influenced because it is uninitialized stack memory that typically contains a leftover stack value from the decoding of a CBOR data item earlier in the data stream.



```
                         switchD_001024de::caseD_3                      XREF[1]:     001024de(j)
0010267d 0f b6 45 d7     MOVZX      EAX,byte ptr [RBP + local_31]
00102681 0f b6 c8        MOVZX      ECX,AL
00102684 48 8d 55 a8     LEA        RDX=>local_60,[RBP + -0x58]
00102688 48 8b 45 88     MOV        RAX,qword ptr [RBP + local_80]
0010268c 89 ce           MOV        ESI,ECX
0010268e 48 89 c7        MOV        RDI,RAX
00102691 e8 31 78        CALL       rtCborDecDynUTF8Str                         undefined rtCborDecDynUTF8Str()
         00 00
00102696 89 45 fc        MOV        dword ptr [RBP + local_c],EAX
00102699 48 8b 55 a8     MOV        RDX=>local_60,qword ptr [RBP + -0x58]
0010269d 48 8b 45 80     MOV        RAX,qword ptr [RBP + local_88]
001026a1 48 89 d6        MOV        RSI,RDX
001026a4 48 89 c7        MOV        RDI,RAX
001026a7 e8 6a 91        CALL       rtJsonEncStringValue                        undefined rtJsonEncStringValue()
```