Talos Vulnerability Report

TALOS-2020-1030

freeDiameter freeDiameterd Denial of Service Vulnerability

JULY 28, 2020

CVE NUMBER

CVE-2020-6098

SUMMARY

An exploitable denial of service vulnerability exists in the freeDiameterd functionality of freeDiameter 1.3.2. A specially crafted Diameter request can trigger a memory corruption resulting in denial-of-service. An attacker can send a malicious packet to trigger this vulnerability.

CONFIRMED VULNERABLE VERSIONS

The versions below were either tested or verified to be vulnerable by Talos or confirmed to be vulnerable by the vendor.

freeDiameter 1.3.2

PRODUCT URLS

freeDiameter - http://www.freediameter.net/

CVSSV3 SCORE

7.5 - CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H

CWE

CWE-191 - Integer Underflow (Wrap or Wraparound)

DETAILS

freeDiameter is an open source implementation of the Diameter protocol specified in RFC3588 (obsoleted by RFC6733)

freeDiameterd is the server implementation of Diameter server protocol. A remote attacker can send a crafted Diameter CER packet with malformed AVP to cause freeDiameterd to crash with a Segmentation fault.

In one of catastrophic test, the AVP payload is filled with 16 null bytes (\x00). Due to a lack of input validation, this results in memory corruption and a crash in freeDiameterd.

AVP Header specification 0 1 2 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 +-+ | AVP Code | +-+ |V M P r r r r r| AVP Length | +-+ | Vendor-ID (opt) | +-+ | Data ... +-+-+-+-+-+-+-+-+

Vulnerable code snippet at messages.c:2112 @ function parsedict_do_avp

```
2103  if (avp->avp_source) {
2104       /* we must copy the data from the source to the internal buffer area */
2105       CHECK_PARAMS( !avp->avp_rawdata  );
2106
2107       avp->avp_rawlen = avp->avp_public.avp_len - GETAVPHDRSZ( avp->avp_public.avp_flags );
2108
2109       if (avp->avp_rawlen) {
2110           CHECK_MALLOC(  avp->avp_rawdata = malloc(avp->avp_rawlen)  );
2111
2112           memcpy(avp->avp_rawdata, avp->avp_source, avp->avp_rawlen);
2113       }
2114
2115          avp->avp_source = NULL;
2116
2117          TRACE_DEBUG(FULL, "Unsupported optional AVP found, raw source data saved in avp_rawdata.");
2118    }
```

Data check at gdb breakpoint (gdb) print *avp $1 = {avp_chain = {chaining = {next = 0x7fff90000b60, prev = 0x7fff90000b60, head = 0x7fff90000b60, o = 0x7fff90000c80}, children = {next = 0x7fff90000ca0, prev = 0x7fff90000ca0, head = 0x7fff90000ca0, o = 0x7fff90000c80}, type = MSG_AVP}, avp_eyec = 288707687, avp_model = 0x0, avp_model_not_found = {mnf_code = 0, mnf_vendor = 0}, avp_public = {avp_code = 0, avp_flags = 0 '\000', avp_len = 0, avp_vendor = 0, avp_value = 0x0}, avp_source = 0x7fff7c000dcc "", avp_rawdata = 0x7ffe63fff010 "", avp_rawlen = 4294967288, avp_storage = {os = {data = 0x0, len = 0}, i32 = 0, i64 = 0, u32 = 0, u64 = 0, f32 = 0, f64 = 0}, avp_mustfreeos = 0}

Related Vulnerabilities:

```
- line #2107, integer underflow of avp->avp_rawlen
  avp->avp_rawlen(4294967288) = avp->avp_public.avp_len(0) - GETAVPHDRSZ(8)
- line #2110, memory exhaustion attack as avp->avp_rawlen integer underflow
- line #2112, memory corruption, as source / dest are NULL pointers and size
  is overflowed.
```

Crash Information

Below is the backtrace when freeDiameterd crashed,

```
Thread 25 "freeDiameterd" received signal SIGSEGV, Segmentation fault.
[Switching to Thread 0x7fffa27fc700 (LWP 13540)]
__memmove_avx_unaligned_erms () at ../sysdeps/x86_64/multiarch/memmove-vec-unaligned-erms.S:384
384     ../sysdeps/x86_64/multiarch/memmove-vec-unaligned-erms.S: No such file or directory.
(gdb) bt
#0  __memmove_avx_unaligned_erms () at ../sysdeps/x86_64/multiarch/memmove-vec-unaligned-erms.S:384
#1  0x00007ffff7cf82c1 in parsedict_do_avp (dict=0x5555555778c0, avp=0x7fff90000c80, mandatory=1, error_info=0x7fffa27fbc00) at
/opt/freeDiameter-1.3.2/libfdproto/messages.c:2112
#2  0x00007ffff7cf93e4 in parsedict_do_chain (dict=0x5555555778c0, head=0x7fff90000b60, mandatory=1, error_info=0x7fffa27fbc00) at
/opt/freeDiameter-1.3.2/libfdproto/messages.c:2268
#3  0x00007ffff7cf98b3 in parsedict_do_msg (dict=0x5555555778c0, msg=0x7fff90000b40, only_hdr=0, error_info=0x7fffa27fbc00) at
/opt/freeDiameter-1.3.2/libfdproto/messages.c:2324
#4  0x00007ffff7cf9b68 in fd_msg_parse_dict (object=0x7fff90000b40, dict=0x5555555778c0, error_info=0x7fffa27fbc00) at /opt/freeDiameter-
1.3.2/libfdproto/messages.c:2353
#5  0x00007ffff7cfb0fa in fd_msg_parse_rules (object=0x7fff90000b40, dict=0x5555555778c0, error_info=0x7fffa27fbc00) at /opt/freeDiameter-
1.3.2/libfdproto/messages.c:2649
#6  0x00007ffff7f5fad8 in client_worker (arg=0x5555555d5540) at /opt/freeDiameter-1.3.2/libfdcore/server.c:222
#7  0x00007ffff7ca9fb7 in start_thread (arg=<optimized out>) at pthread_create.c:486
#8  0x00007ffff7bdb2ef in clone () at ../sysdeps/unix/sysv/linux/x86_64/clone.S:95
```

TIMELINE

2020-04-17 - Vendor Disclosure

2020-07-28 - Public Release

CREDIT

Discovered by Peter Wang of Cisco ASIG.
