IOActive Security Advisory

Title	QNX `ker_msg_sendv` System Call Integer Overflow
Severity	High
Date Discovered	10.30.08
Date Reported	10.30.08
Date Disclosed	10.31.08
Author	Ilja van Sprundel

Affected Products

QNX versions up to 6.4 are vulnerable; the issue was fixed in version 6.4.1.

Description

QNX's `ker_msg_sendv()` system call contains an integer overflow that could lead to heap corruption and, if correctly exploited, system compromise. If it is only partially exploited it could lead to denial of service and kernel panic, effectively shutting down the system.

Callers of ker_msg_sendv() can provide an array of IOVs—the system call reviews and validates them, and sums the total of their lengths, at which point an integer overflow occurs. After that, a similar loop is used copy all the IOVs to a kernel buffer.

```
trunk\services\system\ker\ker_fastmsg.c
int kdecl
ker_msg_sendv(THREAD *act, struct kerargs_msg_sendv *kap) {
...
      if(kap->sparts < 0) {
...
      } else {
            // Multi IOV case
            int len = 0;
            IOV *iov = kap->smsg; ← so we control the iov struct
            int sparts = kap->sparts;
...
            while(sparts) {
                  uintptr_t base, last;

                  len += GETIOVLEN(iov); ← and it's length
                                         ← this can overflow
                  base = (uintptr_t)GETIOVBASE(iov);
                  last = base + GETIOVLEN(iov) - 1;
                  if(((base > last) || !WITHIN_BOUNDRY(base,
last, act->process->boundry_addr)) && (GETIOVLEN(iov) != 0)) {
                  return EFAULT;
                  }
...
            }
...
            if(len <= sizeof(act->args.msbuff.buff)) {
                  int pos = 0;
                  iov = kap->smsg;
                  sparts = kap->sparts;
...
                  while(sparts) {
                        int ilen = GETIOVLEN(iov);
                        __inline_xfer_memcpy(&act-
>args.msbuff.buff[pos], GETIOVBASE(iov), ilen); ← can overflow

                        pos += ilen;
                        iov++;
                        sparts--;
                  }
...
}
```

This vulnerability is likely to be triggered because one could specify a number of IOVs that all point to the same memory, causing the length integer overflow and needing only a small amount of memory to trigger it. The system call in question can be reached by calling the `MsgSendv()` API.

Remediation

This vulnerability was fixed in pr62575, a link to which can be found on:

<http://community.qnx.com/sf/discussion/do/listPosts/projects.core_os/discussion.osrev.topc4767>.