

```
char *format_timespan(char *buf, size_t 1, usec_t t, usec_t accuracy) {
    char *p = buf;
    for (size_t i = 0; i < ELEMENTSOF(table); i++) {</pre>
        size_t n;
        . . . .
        if (!done) {
            k = snprintf(p, 1,
                                 "%s"USEC_FMT"%s",
                                 p > buf ? " " : "",
                                 table[i].suffix);
            t = b;
        }
        n = MIN((size_t) k, 1);
        1 -= n;
        p += n;
    *p = 0;
    return buf;
}
```

The problem of the above code is that  $n = MIN((size_t) k, 1)$ ; can assign the buffer size 1 to n. Then p += n; will cause p to point to one byte after the buffer buf, leading to a buffer overwrite in \*p=0 (an offby-one error).

## Steps to reproduce the problem

To reproduce the buffer overrun, just run the following test code

```
int main() {
   char buf[5];
   char *p;
   usec_t t = 100005;
    usec_t accuracy = 1000;
    p = format_timespan(buf, sizeof(buf), t, accuracy);
    printf("%s\n",p);
    return 0;
}
```

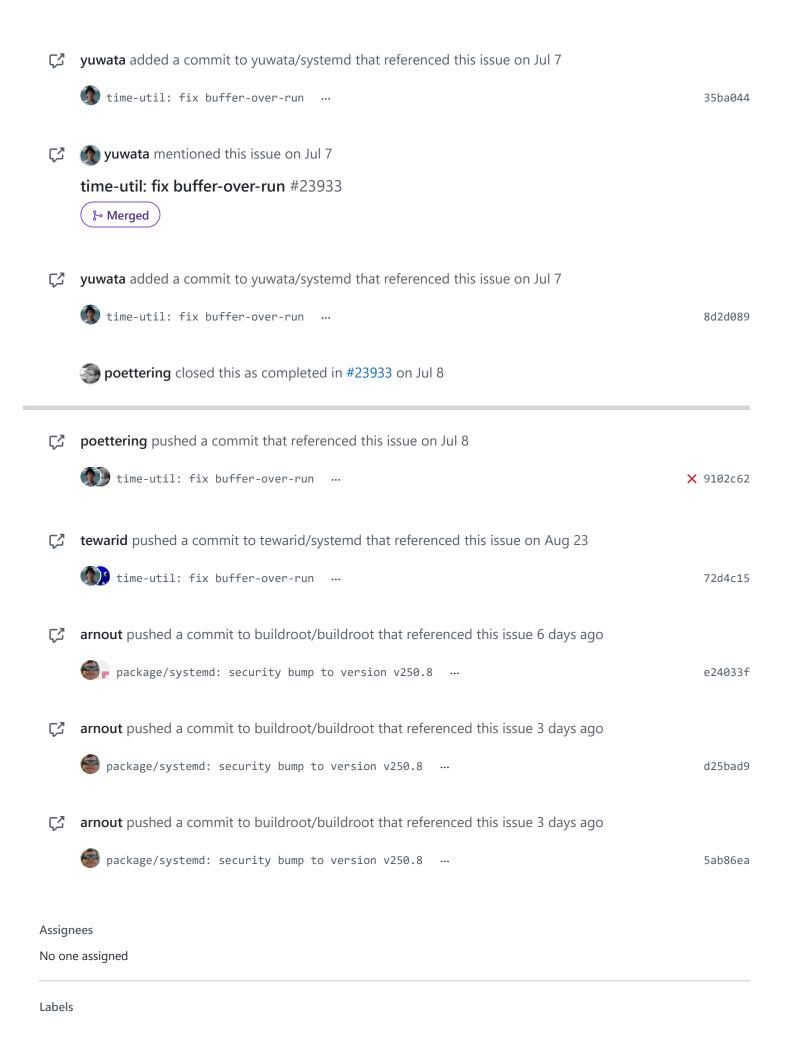
format\_timespan will write to buf[5], which is an error.

## Additional program output to the terminal or log subsystem illustrating the issue

No response









Milestone

No milestone

## Development

Successfully merging a pull request may close this issue.

time-util: fix buffer-over-run yuwata/systemd

## 1 participant

