

TP-Link, Mercury and FAST Router Vulnerability Report

A lot of TP-Link, Mercury and FAST router have a stack overflow issue in `MmtAtePrase` function.

Local users could get remote code execution, this vulnerability currently affects latest WDR series (TP-Link), including WDR5660, WDR7620, WDR7660, WDR7661, etc. We believe there are much more models suffered from this vuln.

Affected Products & Version

TL-WDR7660 2.0.30

Mercury D196G 20200109_2.0.4

Fast FAC1900R 20190827_2.0.2

Vulnerability Description

This vulnerability happen when `MmtAtePrase` receive a malicious string by using `recvfrom` from UDP port 1060.

1. The malicious input(`a1+23060`) is provided by attackers and the length of malicious input is up to 3072 bytes.
2. The first 6 bytes must be "iwpriv" or "wiocli"
3. And then, the malicious input(`a1+23060`) deliver to `MmtAteParse` function.
4. `MmtAteParse` will use `spliter` for truncate the input data. But `spliter` is a `strcpy` like function. No length limitation in `spliter`.
5. However, `MmtAteParse` copy these data to a stack buffer without checking the length and lead to a stack buffer overflow to execute arbitrary code.

```
1 void __fastcall __noreturn tWlanTask(char *a1)
2 {
3     // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL-"+" TO EXPAND]
4
5     while ( 1 )
6     {
7         v63 = 0;
8         do
9         {
10             v67 = 16;
11             memset(v62, 0, sizeof(v62));
12             v2 = *(a1 + 1);
13             memcpy(v50, a1 + 8, sizeof(v50));
14             v65 = 100000;
15             v64 = 0;
16             v3 = select(v2 + 1, v50, 0, 0);
17             if ( v3 <= 0 )
18                 goto LABEL_123;
19             if ( (v50[a1 >> 5] & (1 << (a1 & 0x1F))) != 0 )
20             {
21                 memset(a1 + 23060, 0, 3072);
22                 v65 = recvfrom(a1, 0, 230
23             {
24                 MmtAteParse(0x4240, (a1 +
25                 goto LABEL_116;
26             }
27         }
28     }
```

Up to 3072 bytes can be receive

only check first 6 bytes