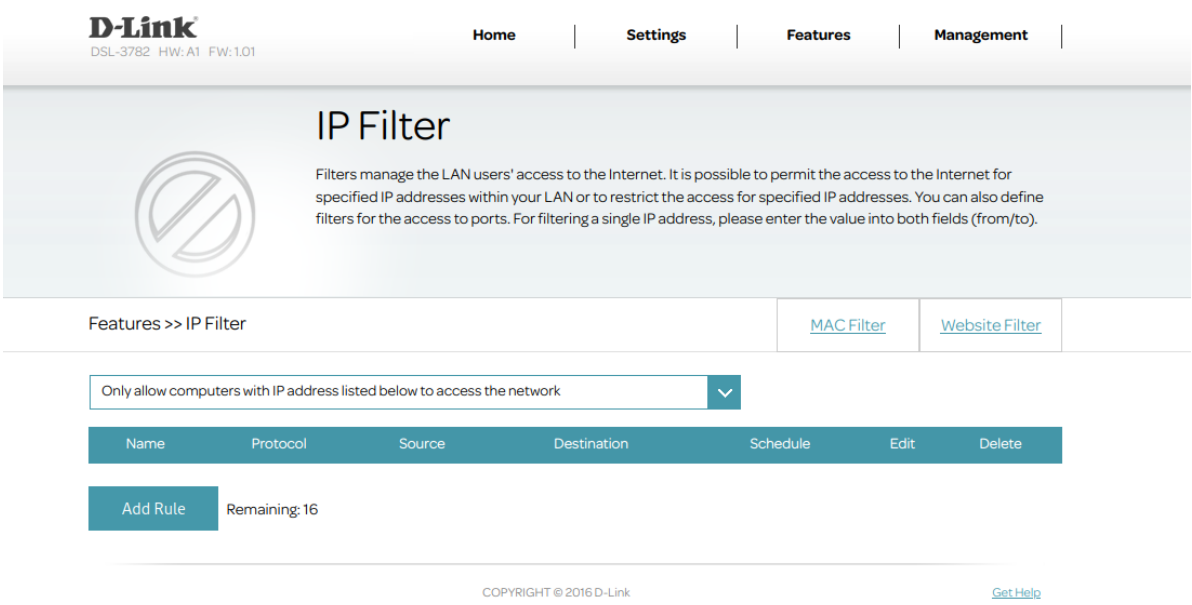
Vendor of the products: D-Link

Affected products: DSL-3782 v1.01

Vulnerability Description

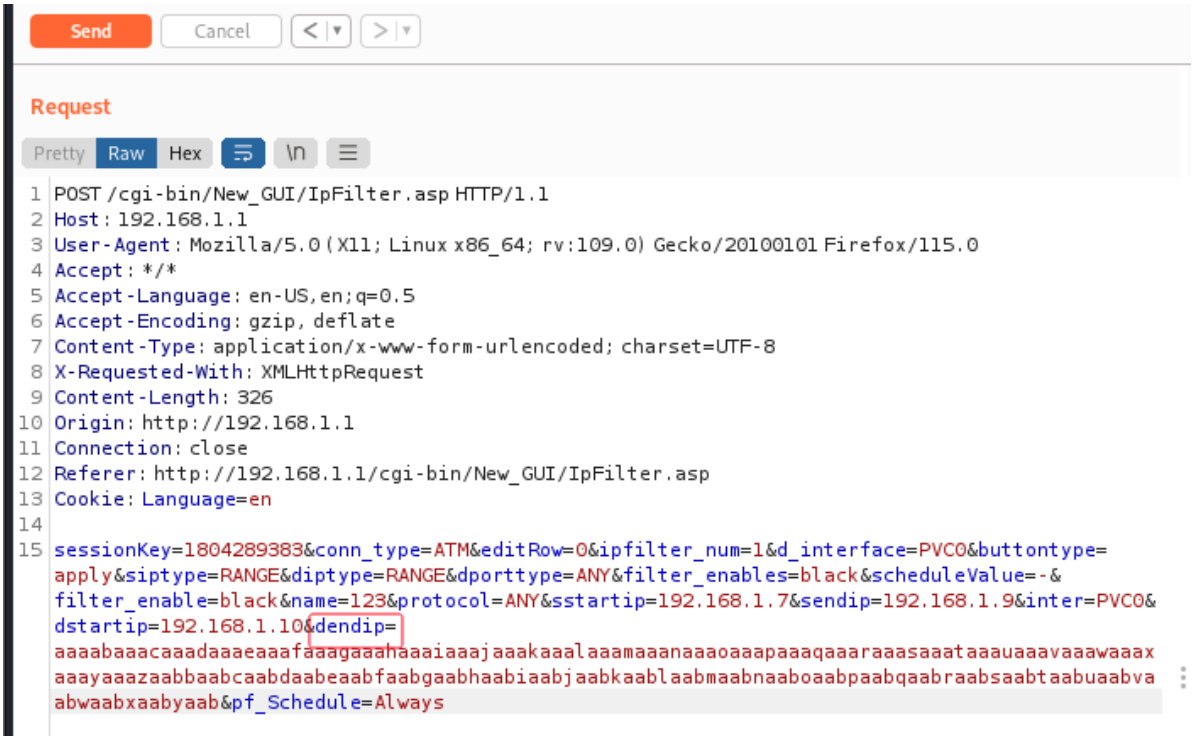
A buffer overflow vulnerability was discovered in D-Link DSL-3782 v1.01, triggered by the sstartip, sendip, dstartip, and dendip parameters. This vulnerability allows attackers to cause a Denial of Service (DoS) via a crafted packet.

The interface that triggers the vulnerability

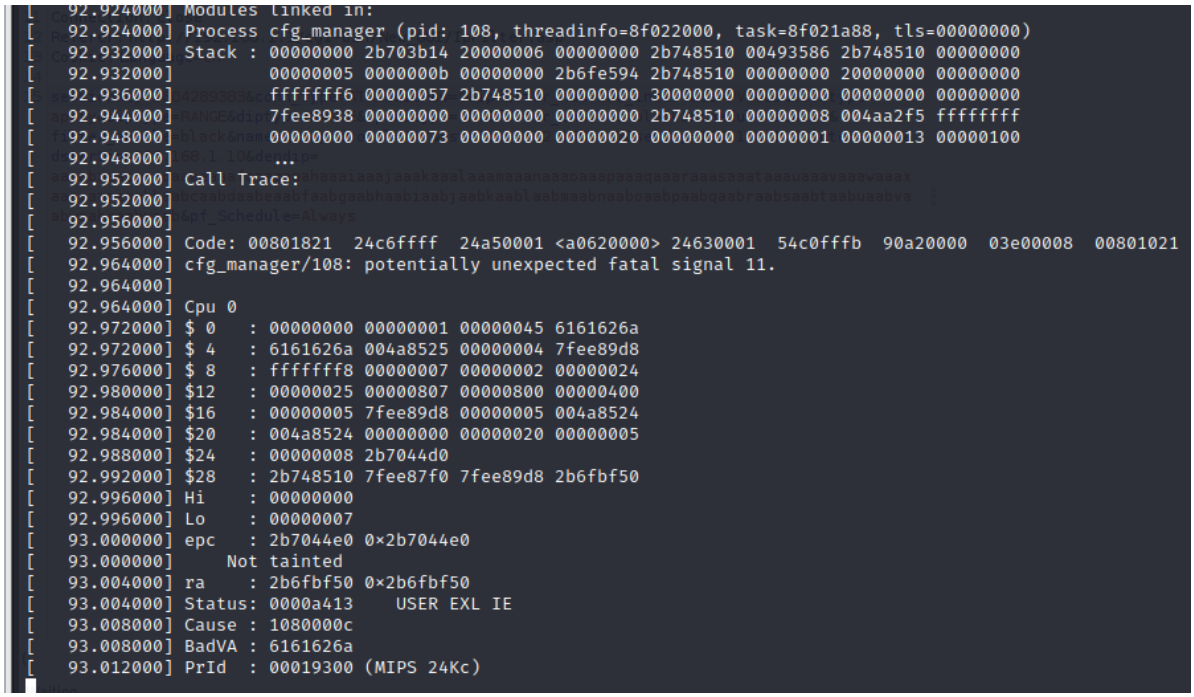


POC

send



You can see that the router has crashed.



Actually, sstartip, sendip, dstartip, and dendip can all trigger the buffer overflow vulnerability.

Code in cfg_manager

By using IDA to analyze cfg_manager, It can be seen that the getAttrValue function is called in IpFilter to retrieve the parameter.

```
16
17  if ( a1 )
18  {
19    if ( a2 )
20    {
21      strcpy(v7, "IpFilter");
22      strcpy(v9, "Common");
23      v7[8] = aIpfilter[8];
24      v8 = 0;
25      v10 = 0;
26      v11 = 0;
27      v12 = 0;
28      v13 = 0;
29      v14 = 0;
30      v15 = 0;
31      memset(v16, 0, 0x81u);
32      if ( !getAttrValue(a1, v7, (int)"Mode", v16) )
33      {
34        if ( strcmp(v16, dword_49A528) )
35        {
36          if ( strcmp(v16, "white") && strcmp(v16, "black") )
37            return 0;
38          memset(v16, 0, 0x81u);
39          if ( getAttrValue(a1, v7, (int)&dword_49D8F8, v16) )
40            return 0;
41          v5 = atoi(v16);
42          if ( v5 < 0 )
43            return 0;
44          v6 = v5 == 0;
45          result = 1;
46          if ( !v6 )
47          {
```

Then it can be seen that the use of strcpy to receive parameters without proper security checks caused the overflow.

```
1  int __fastcall getAttrValue(int a1, char *a2, int a3, char *a4)
2  {
3    int v8; // $s1
4    int v9; // $v0
5    char *v10; // $a2
6    int v11; // $a3
7    int result; // $v0
8    const char *v13; // $a1
9
10   v8 = 0;
11   do
12   {
13     v9 = *a2;
14     v10 = a2;
15     v11 = 0;
16     ++v8;
17     a2 += 16;
18     if ( !v9 )
19       break;
20     a1 = mxmlFindElement(a1, a1, v10, 0, 0, -1);
21   }
22   while ( v8 != 3 );
23   result = -1;
24   if ( a1 )
25   {
26     v13 = (const char *)mxmlElementGetAttr(a1, a3, v10, v11);
27     result = -2;
28     if ( v13 )
29     {
30       strcpy(a4, v13);
31       result = 0;
32     }
33   }
34   return result;
35 }
```