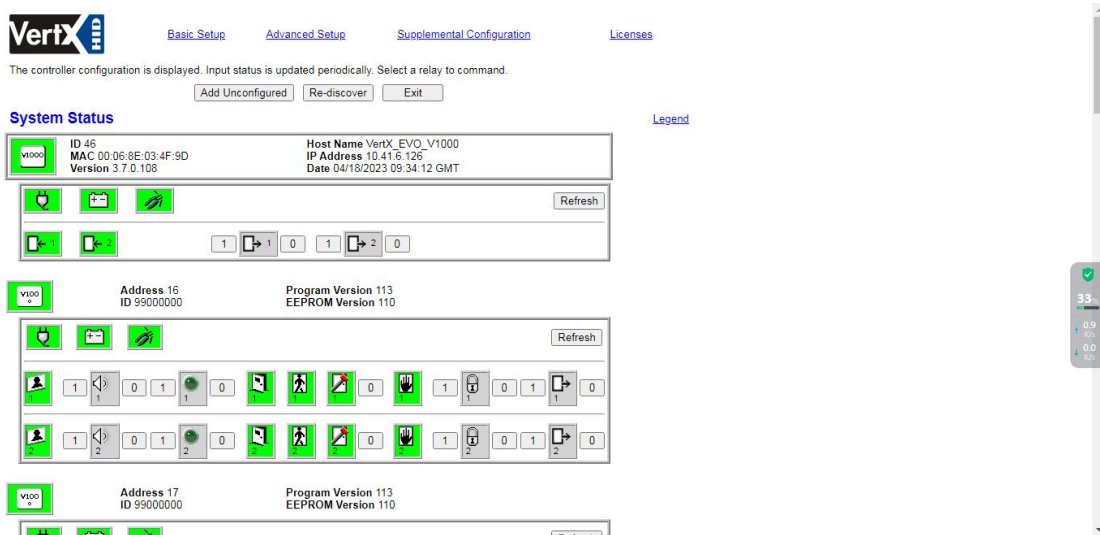


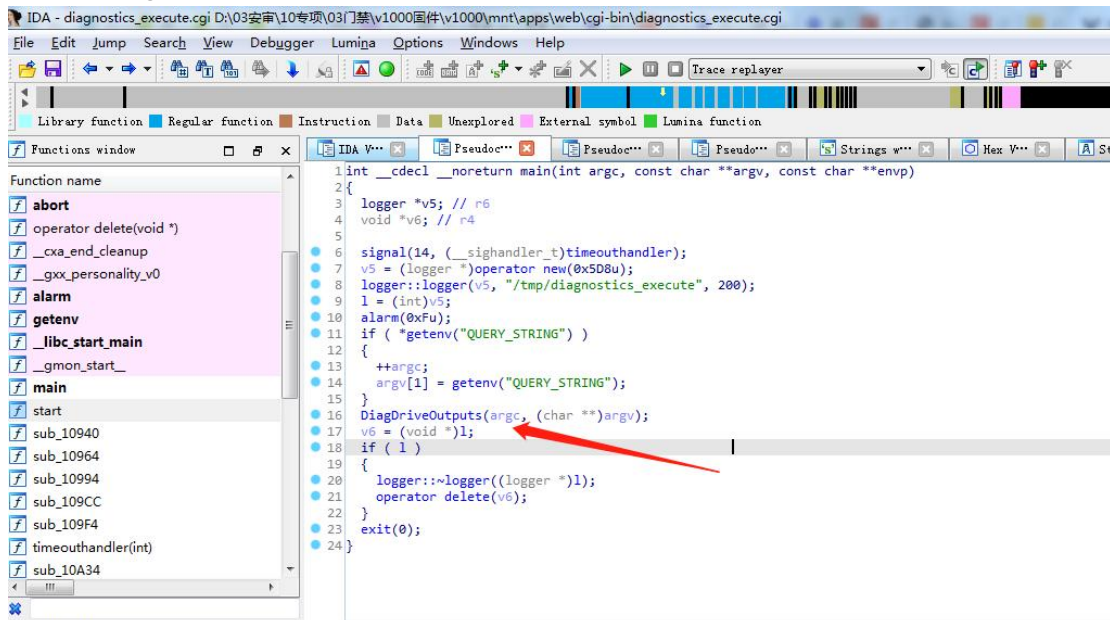
1. BoardType, Description, and Relay parameters overflow

This is the front-end control interface, which can control the opening and closing of the door, etc.

<https://xx.xx.xx.xx/cgi-bin/diagnostics.cgi?Category=status>



The program entry of the backend, the following DiagDriveOutputs is to parse the parameters passed in by the front end, there is a buffer overflow vulnerability in the parsing process of the interface parameters, in which DiagDriveOutputs is implemented by libcgi.so, so the vulnerability exists in libcgi.so



Open Libcgi.so, the buffer sizes of the three parameters of BoardType, Description, and Relay are 20, 20, and 8 respectively, and the function for parsing the three parameters of BoardType, Description, and Relay is getFormValu

```

1 int __fastcall DiagDriveOutputs(int a1, char **a2)
2 {
3     NameValuSet *v2; // r7
4     int v3; // r0
5     int v4; // r5
6     int v5; // r0
7     int v6; // r5
8     int v7; // r4
9     int v9; // r0
10    void ***v10; // r3
11    int v11; // [sp+Ch] [bp-64h] BYREF
12    int v12; // [sp+10h] [bp-60h] BYREF
13    char v13[8]; // [sp+14h] [bp-5Ch] BYREF
14    char v14[20]; // [sp+1Ch] [bp-54h] BYREF
15    char v15[20]; // [sp+30h] [bp-40h] BYREF
16    char s[44]; // [sp+44h] [bp-2Ch] BYREF
17
18    v2 = (NameValuSet *)makeNameValuSet(a1, a2);
19    getFormValu(v2, "ID", &v11);
20    getFormValu(v2, "BoardType", v14);
21    getFormValu(v2, "Description", v15);
22    getFormValu(v2, "Relay", v13);
23    getFormValu(v2, "Action", &v12);
24    if ( v11 == 32 )
25    {
26        silver_fcns::silver_fcns((silver_fcns *)s);
27        v3 = FindOutputBitForRelay((char *)&v1000Relays);
28        v4 = v3;
29        if ( v3 < 0 )
30            syslog(3, "%s: Error - invalid silver bit = %d", "DiagDriveOutputs", v3);
31        if ( v12 == 2 )

```

Open getFormValu, you can see that the length of the buffer is 228, far exceeding the buffer length of the three parameters of BoardType, Description, and Relay, so these are the three buffer overflow points.

```

1 int __fastcall getFormValu(NameValuSet *a1, const char *a2, char *a3)
2 {
3     int v6; // r5
4     char s[228]; // [sp+4h] [bp-E4h] BYREF
5
6     memset(s, 0, 201u);
7     v6 = NameValuSet::find(a1, a2, s);
8     if ( v6 )
9         strcpy(a3, s);
10    return v6;
11 }

```

2. The secondary overflow caused by the overflow of the Description parameter

The maximum length of the Description buffer is 228, and the length of v7 is 20, so strcpy may also cause secondary overflow.

```

16 char v16[20]; // [sp+1Ch] [bp-54h] BYREF
17 char v17[20]; // [sp+30h] [bp-40h] BYREF
18 char s[44]; // [sp+44h] [bp-2Ch] BYREF
19
20 v2 = (NameValuSet *)makeNameValuSet(a1, a2);
21 getFormValu(v2, "ID", &v13);
22 getFormValu(v2, "BoardType", v16);
23 getFormValu(v2, "Description", v17);
24 getFormValu(v2, "Relay", v15);
25 getFormValu(v2, "Action", &v14);
26 if ( v13 == 32 )
27 {
28     silver_fcns::silver_fcns((silver_fcns *)s);
29     v3 = FindOutputBitForRelay((char *)&V100Relays);
30     v4 = v3;
31     if ( v3 < 0 )
32         syslog(3, "%s: Error - invalid silver bit = %d", "DiagDriveOutputs", v3);
33     if ( v14 == 2 )
34         v5 = silver_fcns::set_timed_silver_bit((silver_fcns *)s, v4);
35     else
36         v5 = silver_fcns::set_silver_bit((silver_fcns *)s, v4, v14);
37     v6 = v5;
38     silver_fcns::~silver_fcns((silver_fcns *)s);
39     goto LABEL_8;
40 }
41 if ( !strcmp(v16, "V100") )
42 {
43     v7 = (char *)memset(s, 0, 0x14u);
44     v8 = strcpy(v7, v17);
45     strcat(v8, v15);
46     v9 = FindOutputBitForRelay(V100Relays);
47     if ( !strcmp(v17, "ForcedDoor") )

```