漏洞

Introduce

Hello auditer,here are some Dlink DNS series machine RCE,which can excute code without auth in specify machines.

influent version

- DNS-320 – Version 1.00
- DNS-320LW – Version 1.01.0914.2012
- DNS-325 – Versions 1.01 and 1.02
- DNS-340L – Version 1.08

influent component

These vulnerabilities are belong to account_mgr.cgi ,which cmd parameter(it determined which code brench to run)is:

- cmd=cgi_chg_admin_pw
- cmd=cgi_group_add
- cmd=cgi_group_modify
- cmd=cgi_chg_admin_pw(In fact, I believe there are many more vulnerabilities on this machine. I have only scratched the surface, just 1% of it.)

vulnerable details

First thing first,let's unpack the firmware and get to the file: account_mgr.cgi,then put it into IDA,we can see cgiMain() has so many brench we can go,we need control

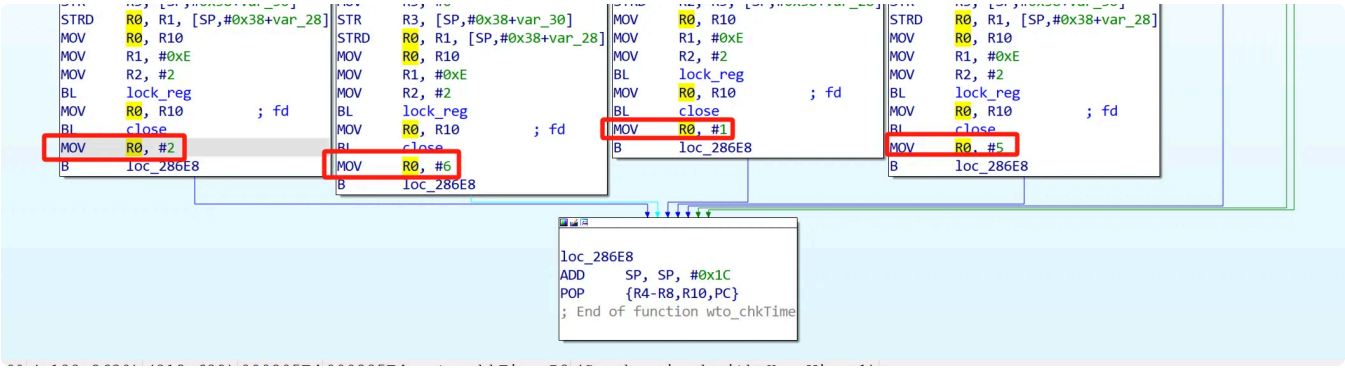
```
1 int cgiMain()
2 {
3   char s1[64]; // [sp+Ch] [bp-50h] BYREF
4   int v2; // [sp+4Ch] [bp-10h]
5
6   v2 = check_login();
7   if ( v2 == 1 )
8   {
9     cgiFormString((int)"cmd", (int)s1, 64);
10    if ( !strcmp(s1, "cgi_open_tree") )
11    {
12      cgi_open_tree();
13    }
14    else if ( !strcmp(s1, "cgi_open_new_folder") )
15    {
16      cgi_open_new_folder();
17    }
18    else if ( !strcmp(s1, "cgi_user_add") )
19    {
20      cgi_user_add();
21    }
22    else if ( !strcmp(s1, "cgi_user_list") )
23    {
24      cgi_user_list();
25    }
26    else if ( !strcmp(s1, "cgi_add_session") )
27    {
```

crontrol to excute which brench

in addition, we don't need to worry about the authentication of `check_login()` (`sub_1CF0C`), because its authentication logic is essentially meaningless. The function `wto_chkTime` will always return a non-zero value(1,2,5,6), which means that even if the user is not logged in, directly accessing the CGI will still cause `check_login` to succeed.

IDA··· Pse··· Sta··· Sta··· Sta··· Sta··· Sta··· Sta···

```
1 int check_login()
2 {
3   int v0; // r5
4   char v2[128]; // [sp+0h] [bp-490h] BYREF
5   char v3[1040]; // [sp+80h] [bp-410h] BYREF
6
7   cgiCookieString("username", v3, 1024);
8   sprintf(v2, "echo '%s' '%s'>/tmp/test", v3, (const char *)cgiRemoteAddr);
9   system(v2);
10  v0 = wto_chkTime(v3, cgiRemoteAddr);
11  sprintf(v2, "echo '%d----------------' >/tmp/timeout", v0);
12  system(v2);
13  return v0;
14 }
```



account_mgr.cgi->cgi_chg_admin_pw

we can submit "cmd=cgi_chg_admin_pw" to get to the vulnerable route

```
186     else if ( !strcmp(cmd_param, "cgi_chk_admin_pw") )
187     {
188       cgi_chk_admin_pw();
189     }
190     else if ( !strcmp(cmd_param, "cgi_chg_admin_pw") )
191     {
192       cgi_chg_admin_pw();
193     }
194     else if ( !strcmp(cmd_param, "cgi_nfs_enable") )
195     {
196       cgi_nfs_enable();
```

```
1  size_t cgi_chg_admin_pw()
2  {
3    int v1[513]; // [sp+0h] [bp-8D4h] BYREF
4    char s[128]; // [sp+806h] [bp-CEh] BYREF
5    char s1[2]; // [sp+886h] [bp-4Eh] BYREF
6    char v4[76]; // [sp+888h] [bp-4Ch] BYREF
7
8    cgiFormString((int)"pw", (int)v4, 64);
9    account_cmd("account -m -u '%s' -p '%s'", "admin", v4);
10   sprintf(s, "snmp_tool -t %d >/dev/null", 1);
11   system(s);
12   syslog(6, "Admin password has been modified.");
13   xml_get_str("/system_mgr/mail/mail_event/name_pwd_enable", 2, s1);
14   if ( !strcmp(s1, "1") )
15   {
16     memset(v1, 0, 0x800u);
17     v1[1] = 7;
18     Lib_Mail_Connect(v1, 15);
19   }
20   LIB_CP_Config_To_MTD(196644);
21   fwrite("Pragma: no-cache\r\nCache-Control: no-cache\r\n", 1u, 0x2Bu, (FILE *)cgiOut);
22   cgiHeaderContentType("text/xml");
23   fwrite("<?xml version=\"1.0\" encoding=\"UTF-8\" ?>\n", 1u, 0x28u, (FILE *)cgiOut);
24   return fwrite("<modify_info><status>1</status></modify_info>\n", 1u, 0x2Eu, (FILE *)cgiOut);
25 }
```

we can submit pw to control the system param

PoC&EXP(sum of the 4-ones vulnerabilities)

the vuln to be tested is determind by vulpoint_template