D-Link DIR-3060 1.11b04 Command Injection

Authored by T Shiomitsu | Site iot-inspector.com

Posted Mar 12, 2021

D-Link DIR-3060 versions 1.11b04 and below suffer from an authenticated command injection vulnerability.

tags | exploit
advisories | CVE-2021-28144
SHA-256 | 934dc62fa5f0b5a818763d562c797ed8d79104a93d069761cc9dcaa5f0408e44 **Download** | **Favorite** | **View**

Related Files

Share This

Like Twee LinkedIn Reddit Digg StumbleUpon

Change Mirror Download

```
IoT Inspector Research Lab Security Advisory IOT-20210311-0
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
              title: Authenticated Command Injection in D-Link DIR-3060 Web
                     Interface
     vendor/product: D-Link DIR-3060 (https://www.dlink.com/)
 vulnerable version: v1.11b04 & Below
      fixed version: v1.11b04 Hotfix 2
         CVE number: CVE-2021-28144
             impact: 8.8 (high) CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H
           reported: 2020-11-27
        publication: 2021-03-11
                 by: T Shiomitsu, IoT Inspector Research Lab
                     https://www.iot-inspector.com/

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Vendor description:
-------------------
D-Link Corporation is a Taiwanese multinational networking equipment
manufacturing corporation. The DIR-3060 (also known as the EXO AC3000 Smart
Mesh Wi-Fi Router) is one of their higher-end home/small business routers.


Vulnerability overview/description:
-----------------------------------
The D-Link DIR-3060 is affected by a post-authentication command injection
vulnerability. Any person who is able to gain authenticated access to a
DIR-3060 would be able to run arbitrary system commands on the device as the
system "admin" user, with root privileges.


Proof of concept:
-----------------
When a SOAP request is made to the SetVirtualServerSettings SOAP endpoint, the
function at 00461918 in prog.cgi is invoked. This function traverses the SOAP
XML request body, stores expected SOAP field values, and takes different paths
depending on the values.

If a request with a non-null LocalIPAddress, Enabled set to “true”, an
InternalPort of “9” and a ProtocolType of “UDP” is sent, the function
CheckArpTables (named by me, based at 0046163c) is invoked.

// ...snip
      iVar5 = strcmp(Enabled,"true");
      if ((((iVar5 == 0) && (LocalIPAddress != (char *)0x0)) &&
          (iVar5 = strcmp(InternalPort,"9"), iVar5 == 0)) &&
         (iVar5 = strcmp(ProtocolType,"UDP"), iVar5 == 0)) {
        local_4154 = local_4154 + 1;
        iVar5 = CheckArpTables(LocalIPAddress, InternalPort, ProtocolType, 0xdc, local_4154);
        if (iVar5 == -1) {
            local_4160 = 0xb;
            goto LAB_00462504;
        }
      }
// ...snip

Interestingly, UDP/9 correlates to the canonical Discard Protocol, which is the
TCP/UDP/IP equivalent of /dev/null.

The CheckArpTables() function attempts to check the device ARP records, by
calling the arp system command and grep’ing the output. However, the user-
controlled value passed as the LocalIPAddress is written directly into the
command line format string with snprint(). This string is then passed directly
to a function called FCGI_popen(), which is a library function imported from
libfcgi.so.

undefined CheckArpTables(char *LocalIPAddress, char *InternalPort, char *ProtocolType, undefined param_4, int
param_5) {
    // ...snip...
    memset(buffer, 0, 0x40);
    // ...snip...
    snprintf(buffer, 0x40, "arp | grep %s | awk \'{printf $4}\'", LocalIPAddress);
    iVar1 = FCGI_popen(buffer, "r");
    // ...snip...
}

We can see in libfcgi.so that FCGI_popen() is essentially only a thin wrapper
around the stdio popen() library function. Arguments passed to FCGI_popen()
get passed directly to popen().


int FCGI_popen(char *param_1, char *param_2)
{
  FILE *__stream;
  int iVar1;
  __stream = popen(param_1,param_2);
  iVar1 = FCGI_OpenFromFILE(__stream);
  if ((__stream != (FILE *)0x0) && (iVar1 == 0)) {
    pclose(__stream);
  }
  return iVar1;
}

Since the LocalIPAddress value is not sanitized or checked in any way, a
crafted command injection string can be passed as the LocalIPAddress, which
will then be written to the arp command format string, and passed (almost)
directly to popen().


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Vulnerable / tested versions:
-----------------------------
DIR-3060 v1.11b04


Solution:
---------
Apply D-Link-supplied patch, v1.11b04 Hotfix 2.


Advisory URL:
-------------
https://www.iot-inspector.com/blog/advisory-d-link-dir-3060/


Vendor contact timeline:
------------------------
2020-11-16: Initial contact made to ipsecure@dlinkcorp.com to request keys for
            encryption.
2020-11-20: No reply received, so follow-up e-mail sent.
2020-11-27: No reply received, so advisory sent by e-mail without encryption.
2021-02-03: No reply received, so follow-up e-mail sent.
```

Follow us on Twitter

Subscribe to an RSS Feed

File Archive: December 2022 <

Su	Mo	Tu	We	Th	Fr
Sa					
				1	2
3					
4	5	6	7	8	9
10					
11	12	13	14	15	16
17					
18	19	20	21	22	23
24					
25	26	27	28	29	30
31					

Top Authors In Last 30 Days

Red Hat 157 files
Ubuntu 76 files
LiquidWorm 23 files
Debian 21 files
nu11secur1ty 11 files
malvuln 11 files
Gentoo 9 files
Google Security Research 8 files
Julien Ahrens 4 files
T. Weber 4 files

File Tags

ActiveX (932)
Advisory (79,754)
Arbitrary (15,694)
BBS (2,859)
Bypass (1,619)
CGI (1,018)
Code Execution (6,926)
Conference (673)
Cracker (840)
CSRF (3,290)
DoS (22,602)
Encryption (2,349)
Exploit (50,359)
File Inclusion (4,165)
File Upload (946)
Firewall (821)
Info Disclosure (2,660)
Intrusion Detection (867)
Java (2,899)
JavaScript (821)
Kernel (6,291)
Local (14,201)
Magazine (586)
Overflow (12,419)
Perl (1,418)
PHP (5,093)
Proof of Concept (2,291)
Protocol (3,435)
Python (1,467)
Remote (30,044)
Root (3,504)
Ruby (594)
Scanner (1,631)
Security Tool (7,777)
Shell (3,103)
Shellcode (1,204)
Sniffer (886)

File Archives

December 2022
November 2022
October 2022
September 2022
August 2022
July 2022
June 2022
May 2022
April 2022
March 2022
February 2022
January 2022
Older

Systems

AIX (426)
Apple (1,926)
BSD (370)
CentOS (55)
Cisco (1,917)
Debian (6,634)
Fedora (1,690)
FreeBSD (1,242)
Gentoo (4,272)
HPUX (878)
iOS (330)
iPhone (108)
IRIX (220)
Juniper (67)
Linux (44,315)
Mac OS X (684)
Mandriva (3,105)
NetBSD (255)
OpenBSD (479)
RedHat (12,469)
Slackware (941)
Solaris (1,607)

```
2021-02-12: No reply received, so inquiry sent using the forms at
            support.dlink.com and eu.dlink.com/uk/en/contact-d-link.
2021-02-17: Response from the US D-Link support team, pointing us towards the
            US-specific D-Link security page.
2021-02-17: Sent e-mail to this new US-specific D-Link security e-mail address.
2021-02-19: Response from a member of the D-Link USA SIRT.
2021-02-19: We request a public key from D-Link USA for transmission of the
            advisory.
2021-02-19: PGP public key is provided by D-Link USA.
2021-02-19: Advisory is sent to D-Link USA with encryption.
2021-02-19: Receipt of advisory is confirmed by D-Link USA SIRT.
2021-02-19: We reply and ask for D-Link USA to keep us updated.
2021-02-20: D-Link “ipsecure” finally answers our e-mail, saying that
            security@dlink.com should be the official e-mail, and the
            ipsecure@dlinkcorp.com e-mail (the only one listed on the main
            D-Link security disclosure page) is only a backup address.
2021-02-22: D-Link USA responds, confirming that the e-mail address listed
            on the main D-Link security page has been changed.
2021-03-02: We e-mail D-Link USA to ask for a status update.
2021-03-02: D-Link USA responds with status update.
2021-03-08: D-Link USA provides patched firmware for testing.
2021-03-08: We respond asking for assigned CVE number.
2021-03-08: D-Link USA notes that they do not apply for, or manage CVE numbers
            related to their own products.
2021-03-08: We apply for a CVE number for this issue.
2021-03-08: D-Link USA publishes public advisory.
2021-03-11: CVE is assigned & IoT Inspector Research Lab publishes advisory.


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

The IoT Inspector Research Lab is an integrated part of IoT Inspector.

IoT Inspector is a platform for automated security analysis and compliance
checks of IoT firmware. Our mission is to secure the Internet of Things. In
order to discover vulnerabilities and vulnerability patterns within IoT devices
and to further enhance automated identification that allows for scalable
detection within IoT Inspector, we conduct excessive security research in the
area of IoT.

Whenever the IoT Inspector Research Lab discovers vulnerabilities in IoT
firmware, we aim to responsibly disclose relevant information to the vendor
of the affected IoT device as well as the general public in a way that
minimizes potential harm and encourages further security analyses of IoT
systems.

You can find our responsible disclosure policy here:
https://www.iot-inspector.com/responsible-disclosure-policy/


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Interested in using IoT Inspector for your research or product?

Mail: research at iot-inspector dot com
Web: https://www.iot-inspector.com
Blog: https://www.iot-inspector.com/blog/
Twitter: https://twitter.com/iotinspector

EOF T Shiomitsu / @2021
```

Login or Register to add favorites

Spoof (2,166)
SQL Injection (16,102)
TCP (2,379)
Trojan (686)
UDP (876)
Virus (662)
Vulnerability (31,136)
Web (9,365)
Whitepaper (3,729)
x86 (946)
XSS (17,494)
Other

SUSE (1,444)
Ubuntu (8,199)
UNIX (9,159)
UnixWare (185)
Windows (6,511)
Other