CVE-2022-38258: DLink DIR 819 LFI And DoS

Introduction

D-Link is one of the leading brands when it comes to manufacturing consumer routers. There's a fair chance that you have a D-Link router back at home. The D-Link DIR 819 model is a very popular router.

In this blog post, I explain how I found a Local-File-Inclusion 0day, now designated as CVE-2022-38258, and escalated it to achieve a Denial-of-Service attack. Though the blog is written with respect to the *DIR-819* model, it should be reproduceable on any router running the same firmware.

Description

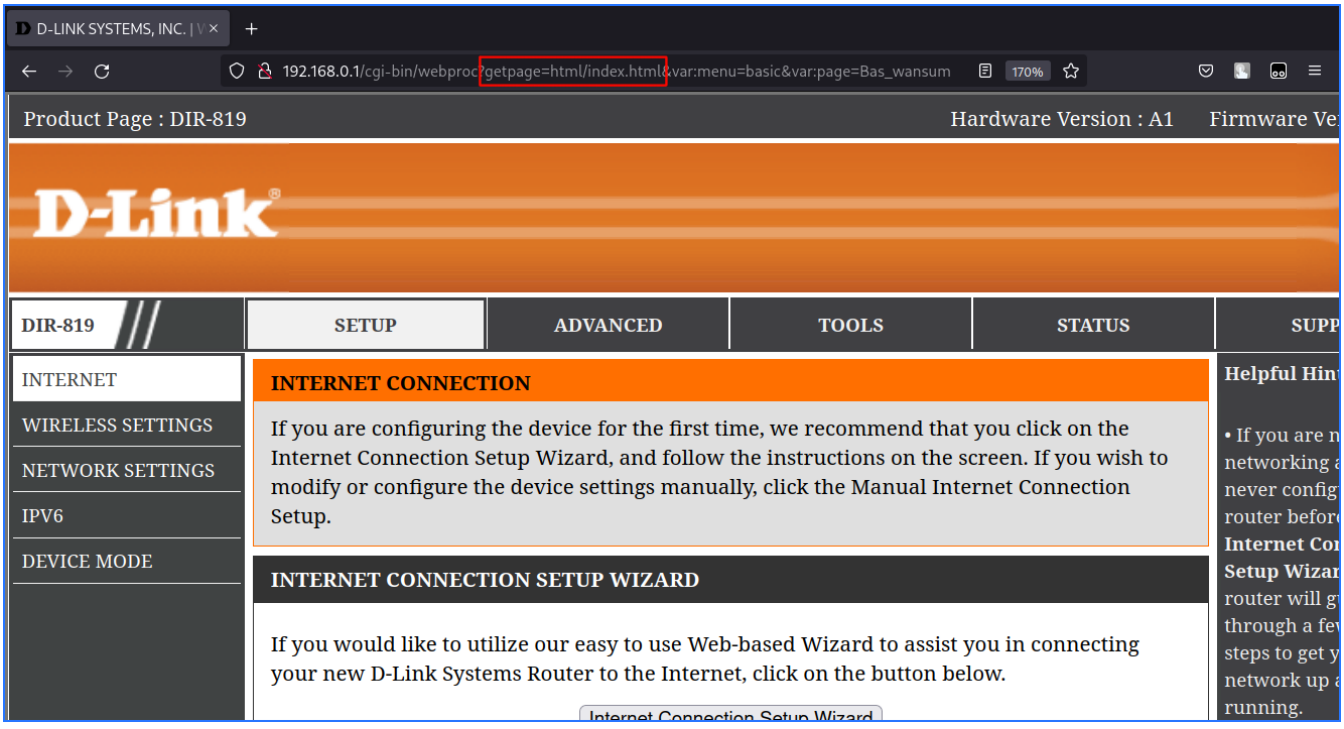
The vulnerability, at its core, is a Local File Inclusion vulnerability which exploits the `webproc` binary. The same vulnerability can then be leveraged to execute a Denial-of-Service attack against the web services.

The vulnerability was tested against the following:

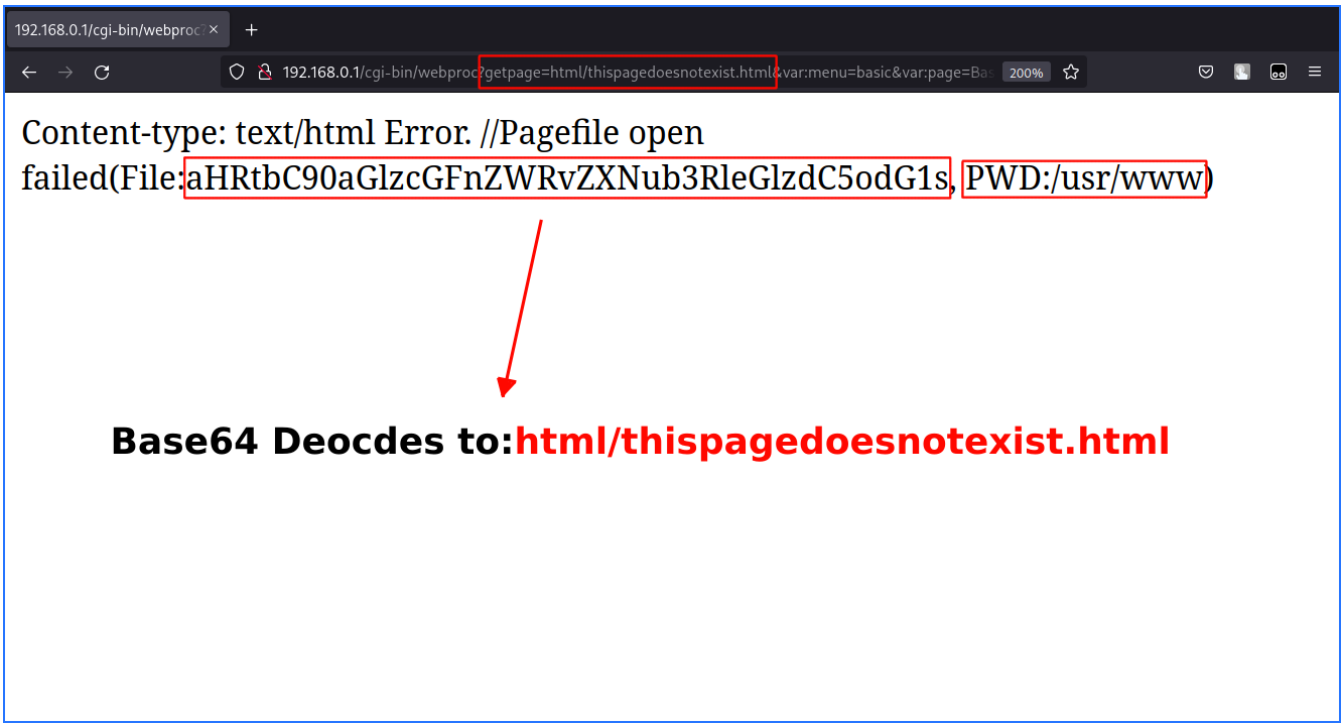
Decription	Value
Model Name	D-Link DIR 819
Firmware Version	V1.06
Hardware Version	A1

Analysis

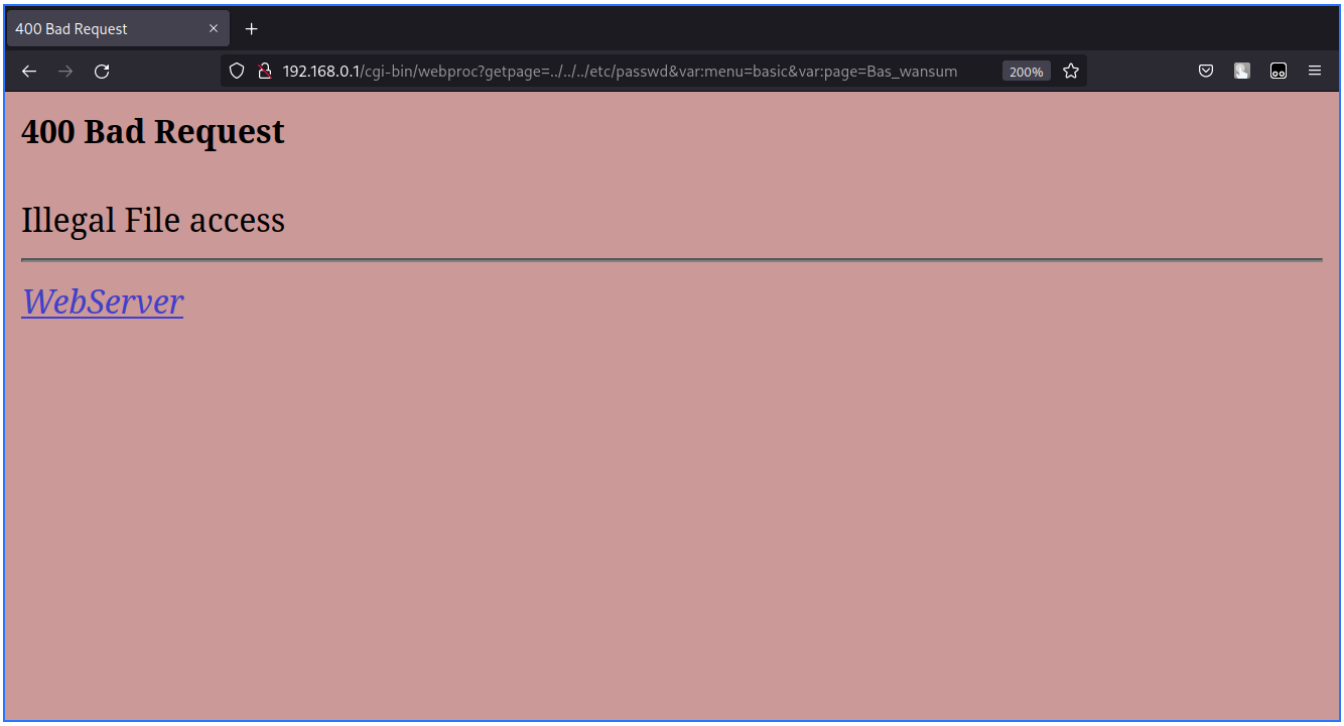
Upon login, the system uses `webproc` to fetch the page to be presented on signup. By default this value is set to `html/index.html` which gives us the following page:



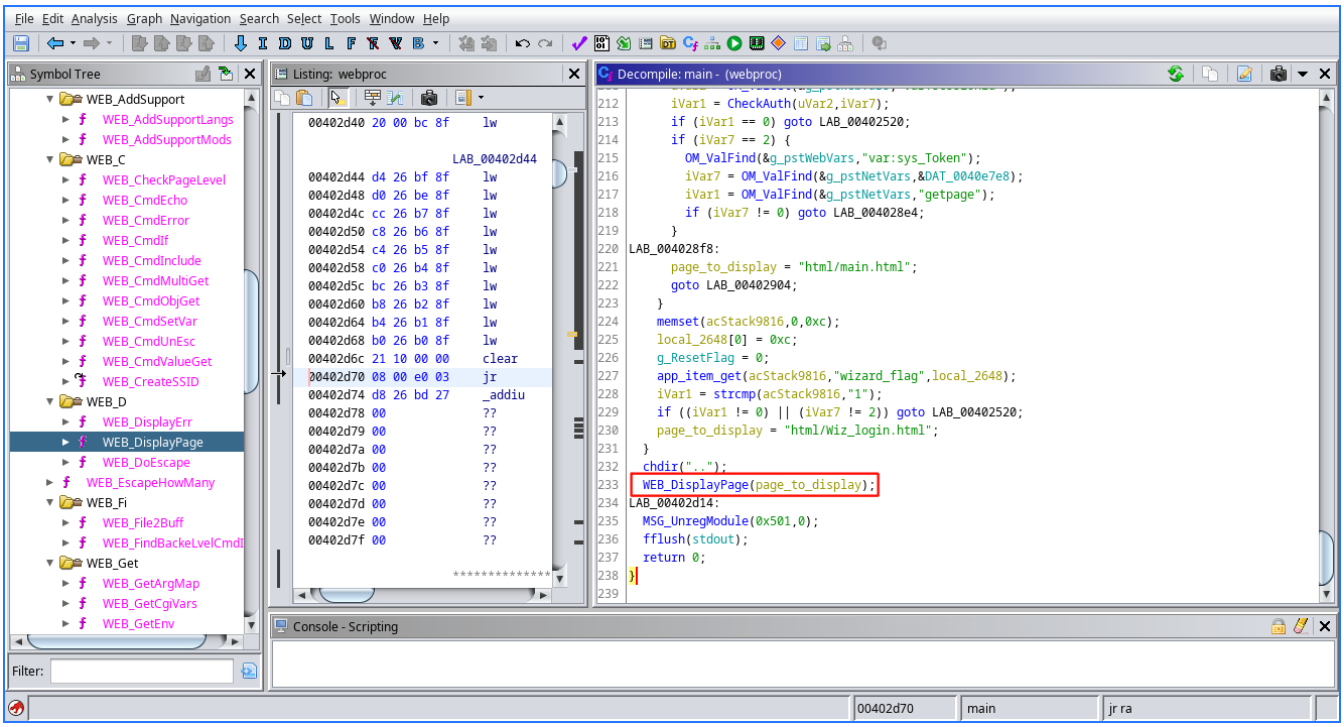
Upon tampering with the `getpage` parameter, we can find that the directory from where the page is being served is `/usr/www`:



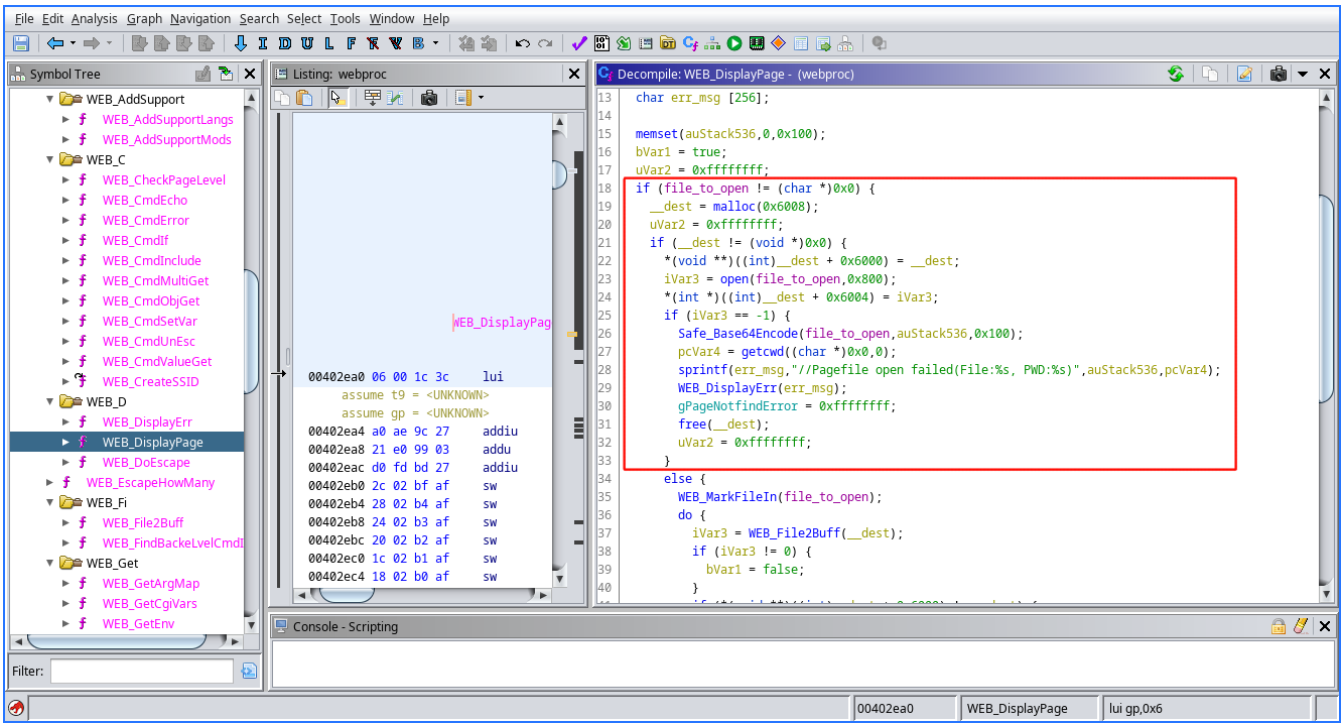
However, at this point, trying to read server files with the Path Traversal payloads returns a **400 Bad Request** error page:



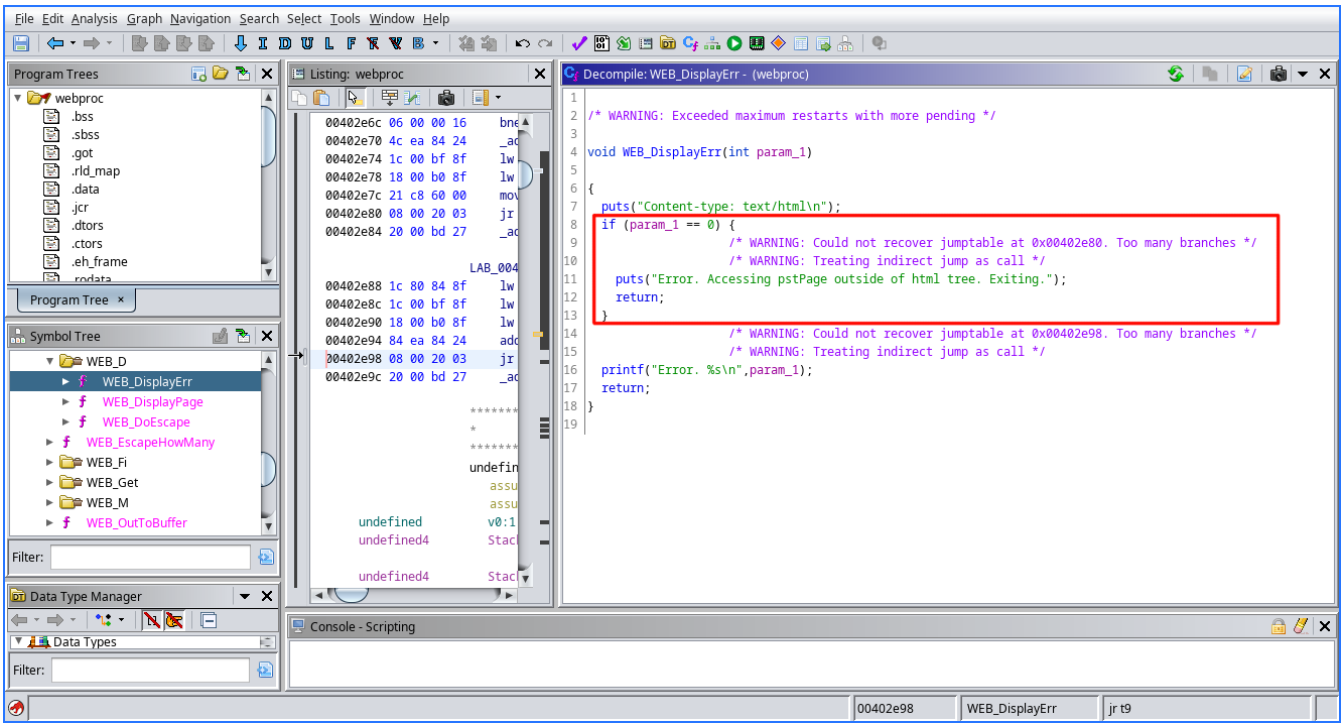
At this point, I decompile extract the firmware with `binwalk` and examine the `webproc` binary by decompiling it with **Ghidra**. Looking at the `main` function, at a first glance, I find the following interesting snippet:



Clearly, the `WEB_DisplayPage` function is responsible for outputting the contents of a page. Examining the function, we see the source of the previously reflected error message:



So, the program checks if the file exists, and if it does, get a handle to the file with `open()` and parses the contents as a string, else returns the Error page by calling `WEB_DisplayErr`. However, as the `WEB_DisplayErr` function states, we can only access pages inside the `html` directory tree(as referenced by the following code):



Thus, with all this information in hand, we can go ahead and craft a request to exploit the vulnerability.

Steps To Reproduce

- Login normally using your credentials
- You should be presented with a similar URL on login:

```
http://192.168.0.1/cgi-bin/webproc?getpage=html/index.html&errorpage=html/m
```

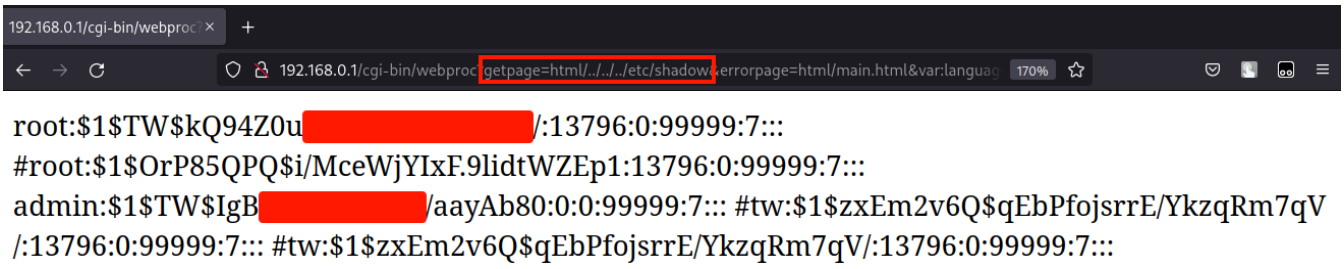
- Replace the `getpage` parameter to:

```
html/%2e%2e/%2e%2e/%2e%2e/etc/shadow
```

- The resulting URL would be similar to:

```
http://192.168.0.1/cgi-bin/webproc?getpage=html/%2e%2e/%2e%2e/%2e%2e/etc/sh
```

- Now, upon hitting enter and making the request, the contents of `/etc/shadow` (or whatever file you requested, will be fetched)



Escalating to Denial of Service

It is possible to launch a Denial of Service attack using the above method. To do this, simply replace the file to read woth `/dev/random` and this shall send the server into a bottomless read operation, effectively causing a Denial of Service attack.

Impact

This vulnerability can allow an attacker to read files on the server, steal credentials and reveal sensitive server side information like log files and such. Also, if an attacker is able to hijack an user session, they can read passwords, credentials etc and can lead to several attacks

References

CVE	Description
CVE-2006-5536	Directory traversal vulnerability in cgi-bin/webcm in D-Link DSL-G624T firmware 3.00B01T01.YA-C.20060616 allows remote attackers to read arbitrary files via a . . (dot dot) in the getpage parameter.
CVE-2006-2337	Directory traversal vulnerability in webcm in the D-Link DSL-G604T Wireless ADSL Router Modem allows remote attackers to read arbitrary files via an absolute path in the getpage parameter

2022-07-31

We were unable to load Disqus. If you are a moderator please see our troubleshooting guide.

Powered by Hugo, theme Anubis.