# CVE-2020-9544: DLink DSL-2640B un-authenticated firmware upgrade (.)

2020-03-05 (.) Gianluca Pacchiella Source (index.md) 1 Comment

**Tags:** CVE (../../../../categories/cve/) router (../../../../categories/router/) security (../../../../categories/security/)

This is a post about CVE-2020-9544 that involves the router DSL-2640B (https://eu.dlink.com/it/it/products/dsl-2640b-adsl-2-wireless-g-router-with-4-port-10-100-switch) by D-Link. I did a simple security assessment on a my old specimen because I changed ISP and this allowed me to change router.

## Vulnerability

This analysis will be very short since the problem is pretty obvious: this is the function that handles the parsing of a HTTP request

```
int request_parse() {
  ...
  __src = fgets(request_line,10000,G_REQUEST_FILE); // [1]
  if (__src == (char *)0x0) {
    __dest = "No request found.";
  }
  else {
    iVar2 = sscanf(request_line,"%[^ ] %[^ ] %[^ ]",method,path,protocol); // [2]
    if (iVar2 == 3) {
        /* parse Authorization, Content-Length and Referer headers */
        ...
    }
  }
  ...
  is_method_POST = strcasecmp(method,"post");
  if (is_method_POST == 0) {                              // [3]
    iVar3 = strcasecmp(path,"/HNAP1/");
    if (iVar3 == 0) {
      /* SOAP related calls */
      ...
      return 0;
    }
    is_path_upload = strcasecmp(path,"/path_to_firmware_upgrade.cgi"); // [4]
    if (is_path_upload == 0) {
      upload_type = 1;
_upload:
      uVar6 = do_upload_pre(G_REQUEST_FILE,parsedContentLength,upload_type);
      return uVar6;                                        // [5]
    }
    local_v0_1832 = strcasecmp(path,"/path_to_settings_upgrade.cgi");  // [6]
    if (local_v0_1832 == 0) {
      upload_type = 2;
      goto _upload;
    }
  }
  ...
}
```

I removed all the code that is not strictly necessary; in `[1]` the first line of the request is read and from that in `[2]` the `method`, `path` and `protocol` are extracted.

After the parsing of a couple of possible headers, the code checks if the request is a `POST` (`[3]`) and in case the path corresponds to a specific string then it uploads the firmware (`[4]`) and the function returns (`[5]`); the same holds for the update of the configuration of the router (`[6]`).

## Disclosure

I think this is a pretty dangerous vulnerability: an attacker with access to the same subnet from which the administration interface is reachable can install her own firmware without any problem! My approach here is to publicly disclose the vulnerability so that anyone that own this router can (hopefully) know that is dangerous to use, since the vendor (that I contacted well before writing this document) has not provided a fix for this.

The path of disclosure with the vendor that I followed:

- 09/10/2019: first message sent to security at dlink.com with the explanation of the problem
- 10/10/2019: reply from D-Link telling me that they will ask R&D for validation

- 28/10/2019: message from me asking for updates
- 30/10/2019: D-Link telling me they are waiting for response from R&D
- 06/02/2020: me asking for a date otherwise I will go public
- 05/03/2020: I have never received feedback and decided to go public

In my opinion this is not acceptable, this is not an obscure vulnerability reachable from an authenticated page via a ROP chain, but a mistake easily preventable with a proper testing phase. How is possible that are necessary months to check if an `if` statement is missing from the source code that you (hopefully) can freely read?

If you really care about security for your home appliance my opinion is that you should change them regularly every 4-5 years and avoid vendor as D-Link that has such history of negligence.

Previous post (../../../02/14/buildroot-getting-started/)

Next post (../../29/exploiting-mips-router/)

## Comments