Talos Vulnerability Report

TALOS-2022-1468

InHand Networks InRouter302 httpd upload.cgi file write vulnerability

MAY 10, 2022

CVE NUMBER

CVE-2022-21809

Summary

A file write vulnerability exists in the httpd upload.cgi functionality of InHand Networks InRouter302 V3.5.4. A specially-crafted HTTP request can lead to arbitrary file upload. An attacker can upload a malicious file to trigger this vulnerability.

Tested Versions

InHand Networks InRouter302 V3.5.4

Product URLs

InRouter302 - https://www.inhandnetworks.com/products/inrouter300.html

CVSSv3 Score

9.9 - CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H

CWE

CWE-377 - Insecure Temporary File

Details

The InRouter302 is an industrial LTE router. It features remote management functionalities and several security protection mechanism, such as: VPN technologies, firewall functionalities, authorization management and several other features.

The InRouter302 offers, once logged, several APIs. One API is called `upload.cgi`. This API allows to upload a file and specify the `type` of file, such as `config`, `modem_upgrade` and `cert_ca`.

The `upload.cgi` API will execute mainly two functions: `upload.cgi_input` that will parse the POST request, and `upload.cgi_output` that will use the parsed input to perform the actual API and return the output, if required. The `upload.cgi_input` function:

```
void upload.cgi_input(char *cgi_filename,uint CONTENT_LENGTH,char *BOUNDARY)

{
  [... several action among which read_buff is filled with the POST content ...]
          filename_provided = strchr(read_buff + 0x26,L'\"');
          if (filename_provided == (char *)0x0) {
            [...]
          }
          *filename_provided = '\0';
          param_value[0] = '\0';
          strlcpy(param_key,read_buff + 0x26,0x80);
          syslog(7,"get var name: %s",param_key);
          filename_provided = strstr(filename_provided + 1,"filename=\"");
          if (filename_provided == (char *)0x0) {
            [...]
          }
          filename_provided = filename_provided + 10;
          filename_end = strchr(filename_provided,L'\"');
          if (filename_end == (char *)0x0) {
            [...]
          }
          *filename_end = '\0';
          pcVar2 = strrchr(filename_provided,L'\\');
[1]
          if (pcVar2 != (char *)0x0) {
            filename_provided = pcVar2 + 1;
          }
          [...]
          snprintf(file_path,0x80,"/tmp/%s",filename_provided);
[2]
          __s = fopen(file_path,"wb");
  [...]
}
```

The two main variables that are going to be parsed, and later used in the `upload.cgi_output`, are `type` and `filename`. The `upload.cgi_input` function is also responsible for creating a temporary file with the content of the provided one. The provided filename, using the `strrchr` function at `[1]`, will be considered, if present, only from the last `\` character in the provided filename. Otherwise the entire provided filename will be used. Then, at `[2]`, the file `/tmp/<provided_filename>` is opened and later filled with the provided content.

Later, in `upload.cgi_output`, based on the `type` variable provided, different actions could be performed. Eventually the temporary file created will be removed. The `upload.cgi_output` function:

```
void upload.cgi_output(void)

{
  [...]

  type = (char *)webcgi_get("type");
  filename = (char *)webcgi_get("filename");
  if ((type == (char *)0x0) || (*type == '\0')) {
    type = "unknown upload type!";
  }
  else {
    [... here it would manage the file based on the type and eventually remove the
temporary file ...]
  }
  syslog(7,type);
LAB_0040ed08:
  if (gl_server_port != 4444) {
    parse_asp("error.jsp");
    return;
  }
  http_api_success = 0;
  return;
}
```

If the `type` variable is not provided, the `upload.cgi` API will not perform any other actions in `upload.cgi_output`. This will result in not deleting the temporary file. Furthermore, at `[2]`, the filename is concatenated without any check or manipulation except for the one performed at `[1]`. This would allow an attacker to perform a path traversal.

The overall impact for these problems will be, for an attacker, to be able to upload and/or overwrite any writable file.

Vendor Response

The vendor has updated their website and uploaded the latest firmware on it. https://inhandnetworks.com/product-security-advisories.html https://www.inhandnetworks.com/products/inrouter300.html#link4

https://www.inhandnetworks.com/upload/attachment/202205/10/InHand-PSA-2022-01.pdf

Timeline

2022-02-25 - Initial vendor contact
2022-03-02 - Vendor Disclosure
2022-05-10 - Public Release

2022-05-10 - Vendor Patch Release

CREDIT

Discovered by Francesco Benvenuto of Cisco Talos.

VULNERABILITY REPORTS PREVIOUS REPORT NEXT REPORT

TALOS-2021-1412 TALOS-2022-1469