Talos Vulnerability Report

TALOS-2022-1575

Robustel R1510 web_server hashFirst denial of service vulnerability

OCTOBER 14, 2022

CVE NUMBER

CVE-2022-35262,CVE-2022-35263,CVE-2022-35267,CVE-2022-35268,CVE-2022-35265,CVE-2022-35270,CVE-2022-35261,CVE-2022-35269,CVE-2022-35266,CVE-2022-35271,CVE-2022-35264

SUMMARY

A denial of service vulnerability exists in the web_server hashFirst functionality of Robustel R1510 3.1.16 and 3.3.0. A specially-crafted network request can lead to denial of service. An attacker can send a sequence of requests to trigger this vulnerability.

CONFIRMED VULNERABLE VERSIONS

The versions below were either tested or verified to be vulnerable by Talos or confirmed to be vulnerable by the vendor.

Robustel R1510 3.1.16
Robustel R1510 3.3.0

PRODUCT URLS

R1510 - https://www.robustel.com/en/product/r1510-industrial-cellular-vpn-router/

CVSSV3 SCORE

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

CWE

CWE-20 - Improper Input Validation

DETAILS

The R1510 is an industrial cellular router. It offers several advanced software features like an innovative use of Open VPN, Cloud management, data over-use guard, smart reboot and others.

The R1510 uses the embedthis's GoAhead library for its web server. More specifically it uses version 4.0.0. This version is vulnerable to a denial of service attack, which would result in a crash of the web server. The library's `hashFirst` function that is used to traverse a structure of hash keys:

```
WebsKey *hashFirst(WebsHash sd)
{
    HashTable   *tp;
    WebsKey     *sp;
    int         i;

    assert(0 <= sd && sd < symMax);
[1]
    tp = sym[sd];
[2]
    assert(tp);

    /*
        Find the first symbol in the hashtable and return a pointer to it.
     */
    for (i = 0; i < tp->size; i++) {
        if ((sp = tp->hash_table[i]) != 0) {
            return sp;
        }
    }
    return 0;
}
```

This function receives as argument a `WebsHash` value, this is practically an identifier for a specific hash table. The function, at `[1]`, checks using `assert` if the identifier is in a valid range value. The function `assert` is not the standard one that will raise a `SIGABRT`, but one defined in the library:

```
extern void assert(bool cond);
#elif ME_GOAHEAD_DEBUG
    #define assert(C)       if (C) ; else assertError(WEBS_L, "%s", #C)
    PUBLIC void assertError(WEBS_ARGS_DEC, char *fmt, ...);
#else
    #define assert(C)       if (1) ; else {}
#endif
```

Essentially this function will do nothing, possibly calling `assertError`, which will print an error message and continue the execution. So if the `sd` value is not in a valid range, it will reach the instruction at `[2]` anyway. This instruction will use the `sd` value as index of an array. This can cause a segmentation fault, which can lead to the termination of the process.

For instance a pattern that can be found in several `web_server`'s APIs is the following:

```
    is_post = scaselessmatch(webs->method,"POST");
    if (is_post != 0) {
      [...]
      webs_files = hashFirst(webs->files);
  [3]
      [...]
```

The API checks if the request is a POST. If so, at `[3]` the files WebsHash is used as argument for the `hashFirst` function. But, if the request didn't include any files, the `webs->files` value would be -1. Then at `[2]` the value -1 is used to access the array. This can cause a denial of service.

Following the APIs identified as vulnerable to this problem.

CVE-2022-35261 - /action/import_authorized_keys/ denial of service

This denial of service is in the `/action/import_authorized_keys/` API.

CVE-2022-35262 - /action/import_xml_file/ denial of service

This denial of service is in the `/action/import_xml_file/` API.

CVE-2022-35263 - /action/import_file/ denial of service

This denial of service is in the `/action/import_file/` API.

CVE-2022-35264 - /action/import_aaa_cert_file/ denial of service

This denial of service is in the `/action/import_aaa_cert_file/` API.

CVE-2022-35265 - /action/import_nodejs_app/ denial of service

This denial of service is in the `/action/import_nodejs_app/` API.

CVE-2022-35266 - /action/import_firmware/ denial of service

This denial of service is in the `/action/import_firmware/` API.

CVE-2022-35267 - /action/import_https_cert_file/ denial of service

This denial of service is in the `/action/import_https_cert_file/` API.

CVE-2022-35268 - /action/import_sdk_file/ denial of service

This denial of service is in the `/action/import_sdk_file/` API.

CVE-2022-35269 - /action/import_e2c_json_file/ denial of service

This denial of service is in the `/action/import_e2c_json_file/` API.

CVE-2022-35270 - /action/import_wireguard_cert_file/ denial of service

This denial of service is in the `/action/import_wireguard_cert_file/` API.

CVE-2022-35271 - /action/import_cert_file/ denial of service

This denial of service is in the `/action/import_cert_file/` API.

TIMELINE

2022-07-13 - Vendor Disclosure

2022-10-14 - Public Release

CREDIT

Discovered by Francesco Benvenuto of Cisco Talos.

VULNERABILITY REPORTS

PREVIOUS REPORT

TALOS-2022-1517

NEXT REPORT

TALOS-2022-1576