Talos Vulnerability Report

TALOS-2022-1579

Robustel R1510 web_server /ajax/remove/ directory traversal vulnerability

OCTOBER 14, 2022

CVE NUMBER

CVE-2022-33897

SUMMARY

A directory traversal vulnerability exists in the web_server /ajax/remove/ functionality of Robustel R1510 3.1.16. A specially-crafted network request can lead to arbitrary file deletion. 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

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:H/A:N

CWE

CWE-22 - Improper Limitation of a Pathname to a Restricted Directory (‘Path Traversal’)

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's web_server has an API that is used to remove a specific file from a preset of folders. The `/ajax/remove/` API expects the `file_name` parameters used to specify the file of interest and other two parameters to choose between a list of possible folders, from which it is possible to remove files.

Here is a relevant portion of the `/ajax/remove/` API:

```
[...]
file_name = (char *)websGetVar(webs,"file_name",0);
generated_filepath._0_4 = 0;
memset(generated_filepath + 4,0,0xffc);
if ((file_name == (char *)0x0) || (is_true = string_inject_verify(file_name,1),
is_true != 0)) {        [1]
  pcVar2 = "illegality arg\n";
}
else {
  is_true = create_sdk_path(webs,file_name,0xffffffff,generated_filepath);
[2]
  if (is_true == 0) {
    exceve_shell_cmd[0] = "rm";
    exceve_shell_cmd[1] = "-rf";
    exceve_shell_cmd[3] = (char *)0x0;
    exceve_shell_cmd[2] = (char *)&generated_filepath;
    is_true = _eval(exceve_shell_cmd,0,0,0);
[3]
    [...]
```

At `[1]` the `file_name` variable is checked against a list of characters that could cause a command injection. If the `file_name` passes the check, then, at `[2]`, the function `create_sdk_path` will be called to create an absolute path. This function will use request's other two parameters to create a full absolute path, using the `file_name` parameter as the file name. The created path will then be used, at `[3]`, to execute the command `rm -rf <created_path>`. From the fetch of the `file_name` variable to the execution function at `[3]`, no check for a path traversal is performed. Because of the missing check, this API is vulnerable to a path traversal vulnerability. This can lead to arbitrary file deletion.

TIMELINE

2022-07-13 - Vendor Disclosure
2022-10-14 - Public Release

CREDIT

Discovered by Francesco Benvenuto of Cisco Talos.