Talos Vulnerability Report

TALOS-2022-1577

Robustel R1510 js_package install OS command injection vulnerability

OCTOBER 14, 2022

CVE NUMBER

CVE-2022-33150

SUMMARY

An OS command injection vulnerability exists in the js_package install functionality of Robustel R1510 3.1.16. A specially-crafted network request can lead to arbitrary command execution. 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

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

CWE

CWE-78 - Improper Neutralization of Special Elements used in an OS Command (‘OS Command Injection’)

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 provides a functionality to add Nodejs app. This functionality is a three step process. First the `/action/import_nodejs_app/` API is used to actually import the Nodejs app into the router. Then the `/ajax/system_nodejs_app_upgrade_start/` API is used to start the installation process. Indeed, this API will finally call the `js_package` package binary with the command `install`.

Here is the relevant code of the `/action/import_nodejs_app/` API:

```
void /action/import_nodejs_app/(Webs *webs)
{
  [...]
        next_file = (WebsKey *)hashFirst(webs->files);
        pcVar5 = file_path;
        for (; next_file != (WebsKey *)0x0; next_file = (WebsKey *)hashNext(webs-
>files,next_file))
        {
          WebsUpload = (next_file->content).WebsUpload;
          [...]
          file_path = (char *)sfmt("%s/%s","/tmp/upload",WebsUpload-
>clientFilename);
          pcVar3 = (char *)rename(WebsUpload->filename,file_path);
          if (pcVar3 == (char *)0xffffffff) {
            [...]
          }
          else {
            FILENAME = string_clone(file_path);
          }
        }
      }
      [...]
```

This function will import the request's uploaded file and set the global `FILENAME` variable with the value `/tmp/upload/<WebsUpload->clientFilename>` where `WebsUpload->clientFilename` is the filename provided in the multipart request.

To start the installation process of the uploaded app, the `/ajax/system_nodejs_app_upgrade_start/` API must be called.

Here is the relevant part of the `/ajax/system_nodejs_app_upgrade_start/` API:

```
undefined4 /ajax/system_nodejs_app_upgrade_start/(Webs *webs)
{
  [...]
  command_array[0] = "js_package";
  command_array[1] = "install";
  command_array[2] = FILENAME;
  command_array[3] = (char *)0x0;
  _eval(command_array,0,0,&pid);
  [...]
}
```

This function, eventually, will execute the `js_package install <FILENAME>` command. The `js_package` binary is responsible for actually performing the installation.

The `js_package` code:

```
int js_package(int argc,undefined4 *argv)
{
  [...]
  command = (char *)argv[1];
  is_install = strncmp(command,"install",7);
  if (is_install == 0) {
    [...]
    filename = (char *)argv[2];
    extension = strrchr(filename,L'.');
    if (extension == (char *)0x0) {
        [...]
    }
    else {
      extension_after_dot = extension + 1;
      [...]
      is_extension_tar = strncmp(extension_after_dot,"tar",3);
      if (is_extension_tar == 0) {
        command = "tar xf %s -C %s";
        [...]
        sysprintf(command,filename,"/app/node-js-app/tmp");
[1]
        [...]
}
```

This function, if the `FILENAME` ends with `.tar`, will perform the `sysprintf` function. This function will execute the `vsnprintf` function with the provided arguments, then use the output as argument for the `system` function. So, at `[1]`, the command `tar xf <FILENAME> -C /app/node-js-app/tmp` will be executed. Because `filename` is the same `FILENAME` specified in `/action/import_nodejs_app/`, this function is vulnerable to a command injection.

TIMELINE

2022-07-13 - Vendor Disclosure

2022-10-14 - Public Release

CREDIT

Discovered by Francesco Benvenuto of Cisco Talos.

VULNERABILITY REPORTS

PREVIOUS REPORT

TALOS-2022-1579

NEXT REPORT

TALOS-2022-1578