Talos Vulnerability Report

TALOS-2021-1425

Reolink RLC-410W cgiserver.cgi cgi_check_ability improper access control vulnerabilities

JANUARY 26, 2022

CVE NUMBER

CVE-2021-40413, CVE-2021-40414, CVE-2021-40415,CVE-2021-40416

Summary

Multiple incorrect default permissions vulnerabilities exist in the cgiserver.cgi cgi_check_ability functionality of reolink RLC-410W v3.0.0.136_20121102. A specially-crafted HTTP request can lead to denial of service. An attacker can send an HTTP request to trigger this vulnerability.

Tested Versions

Reolink RLC-410W v3.0.0.136_20121102

Product URLs

RLC-410W - https://reolink.com/us/product/rlc-410w/

CVSSv3 Score

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

CWE

CWE-284 - Improper Access Control

Details

The Reolink RLC-410W is a WiFi security camera. The camera includes motion detection functionalities and various methods to save the recordings.

The RLC-410W offers several APIs. Each one requires a specific user permission to be executed. An error in the permission check exists that would allow unprivileged users to execute privileged APIs.

The permission is specified using a numeric value, allegedly. Only three bits are used:

```
G S A
2 1 0
```

The bit 2 (value 4) is, allegedly, used to permit the majority of `Get` APIs. The bit 1 (value 2) is, allegedly, used to permit the majority of `Set` APIs. The bit 0 (value 1) is, allegedly, used to permit the most critical APIs, including some `Get` and `Set`. For example, APIs that required the bit 0 set are: `UpgradePrepare`, `Upgrade`, `Reboot`, `Shutdown` and others.

The permission is tangled with the user session. Indeed a table exists in the context of the user session that shows each API category with the corresponding user permission.

If the API request is performed by a logged-in user, the permission is checked by the `cgi_check_ability` function:

```
undefined4 cgi_check_ability(API_command API_command,cgi_session *session,int channel)

  {
    [...]

    ability_struct = session->ability_struct;
    if (API_command == Login) {
      return 0;
    }
    command_struct = cgi_find_cmd_table(API_command);
    if (command_struct == (command_struct *)0x0) {
      return not support;
    }
    if (false) {
  switchD_0043a174_caseD_b:
      session_ability_ = 7;                                                    [1]
      goto LAB_0043a314;
    }
    switch(API_command) {
    case GetDevInfo:
      session_ability_ = ability_struct->GetDevInfo;
      break;
    [... other API cases ...]
    default:
      goto switchD_0043a174_caseD_b;                                           [2]
    [... other API cases ...]
      break;
    case GetCloud:
    case SetCloud:
    case GetCloudSchedule:
    case SetCloudSchedule:
      session_ability_ = ability_struct->Cloud;
      break;
    case GetPowerLed:
    case SetPowerLed:
      session_ability_ = ability_struct->PowerLed;
    }
  LAB_0043a314:
    uVar2 = ability error;
    if ((session_ability_ & command_struct->ability_cmd) != 0) {             [3]
      uVar2 = 0;
    }
    return uVar2;
  }
```

The `cgi_check_ability` checks, given the requested `API_command`, if the user permission satisfies the API permission. If so the API is executed. If the API requested is not within the switch cases, then the default case, at `[2]`, is performed. The check of the permission is performed at `[3]` using a logical `AND` operator between the user permission and the required command permission. It means that, if the permission guaranteed to the user is `7`, like for the default case at `[2]`, all the APIs with a permission value are allowed.

The following APIs do not have a specific case within the switch:

```
{'Login', 'HeartBeat', 'GetMdState', 'GetHddInfo', 'Unknown', 'Playback', 'UpgradePrepare', 'Format', 'SetMdAlarm', 'GetWifiSignal',
'GetAbility', 'GetMdAlarm', 'Logout'}
```

An authenticated user would already be able to access the above `Get` APIs, but APIs like `UpgradePrepare`, `Format` and `SetMdAlarm` should not be executable by a non-admin user. Their impact is described below in dedicated sections.

Note that, while this issue requires a logged-in user, it's possible to use TALOS-2021-1420 to perform these API calls without authentication. In this case, the actual chained CVSS score would be 8.6 - CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:H and would include the `Get` APIs, since an authenticated user is not necessary.

CVE-2021-40413 - UpgradePrepare

The `UpgradePrepare` is the API that checks if a provided filename identifies a new version of the RLC-410W firmware. If the version is new, it would be possible, allegedly, to later on perform the `Upgrade`. Here is the relevant part of the `UpgradePrepare` API code:

```
undefined4 UpgradePrepare(cgi_session *session,cgi_cmd *cgi_cmd)

{
  [...]
          current_time = cgi_tick_sec_get();
          cgi_obj->time_of_UpgradePrepare = current_time;                  [4]
          cgi_cmd->API_processing_status = 3;
          return 0;

  [...]
}
```

Eventually, in the `UpgradePrepare` function, at `[4]` the `time_of_UpgradePrepare` field will be set and later on checked in `cgi_proc`.

The relevant part of the `cgi_proc` function:

```
undefined4 cgi_proc(c_cgiserver_obj *cgi)

{
  [...]
      current_time = cgi_tick_sec_get();
      if (300 < current_time - cgi->time_of_UpgradePrepare) {                                    [5]
        if (cgi->time_of_UpgradePrepare == 0) {
          return 0;
        }
        cgi_log(1,0,"[%s,%d]",
                "/home/lgb/ipc_v1_ver/20201211/ipc_20200324_V1/product/cgiserver/src/cgi_server.cpp"
                ,0x1e5);
        cgi_log(0,0,"upgrade out of time!!!!!");
        cgi->reboot_proc_status = 1;
        cgi->is_upgrade_started = 0;
      }
      uVar1 = 0;
    }
  [...]
}
```

At `[5]` it is shown that, if the `Upgrade` is not executed within 5 minutes of the completion of `UpgradePrepare`, the device will reboot. In `cgi_check_ability` the `UpgradePrepare` API does not have a specific case, the user permission will default to 7. This will give non-administrative users the possibility to reboot the device. Furthermore many other services are going to shut down just after the `UpgradePrepare` request is performed, making the recording and other services unavailable.

CVE-2021-40414 - SetMdAlarm

The `SetMdAlarm` API sets the movement detection parameters, giving the ability to set the sensitivity of the camera per a range of hours, and which of the camera spaces to ignore when considering movement detection. In `cgi_check_ability` the `SetMdAlarm` API does not have a specific case, the user permission will default to 7. This will give non-administrative users the possibility to change the movement detection parameters.

CVE-2021-40415 - Format

The `Format` API formats the SD card, deleting all the recordings in it. After the SD card is formatted, the device will reboot. Because in `cgi_check_ability` the `Format` API does not have a specific case, the user permission will default to 7. This will give non-administrative users the possibility to format the SD card and reboot the device.

CVE-2021-40416 - Get APIs

All the `Get` APIs that are not included in `cgi_check_ability` are already executable by any logged-in users. In TALOS-2021-1420, the`Get` APIs are not included and that is an issue. This will allow `Get` APIs without authentication.

Timeline

2021-12-06 - Vendor Disclosure
2022-01-19 - Vendor Patched
2022-01-26 - Public Release

CREDIT

Discovered by Francesco Benvenuto of Cisco Talos.
