

Tenda AC1206 韌體模擬與漏洞挖掘

研究簡介

IoT 設備隨處可見，如日常使用之電子門鎖、智慧手錶與路由器皆在其中。因此，IoT 設備之安全性便十分重要。然而，許多 IoT 設備的 Code Quality 不優、存在許多 Comand Injection、Buffer Overflow 等漏洞。本研究透過拆解 Tenda AC1206 路由器韌體並分析攻擊面，挖掘其中的漏洞，最終在 Tenda AC1206 路由器上找到數個漏洞並取得 CVE。這篇研究會介紹我們使用的方式，包含韌體拆解與模擬、研究的攻擊面架構簡介，以及挖掘到的漏洞 CVE-2024-53621 與 CVE-2025-50813

韌體拆解

要開始對 IoT 設備進行漏洞研究，我們需要取得 source code 或執行檔以進行進一步的分析，部分廠商會在官方網站上提供韌體下載。然而，這些韌體並非執行檔，需要經過一些方法處理才能取得執行檔開始分析。對於非加密任體，binwalk 可以很好的分析韌體的檔案結構並拆解韌體，取得包含執行檔的檔案系統，拆解後即可使用 IDA / Ghidra 等工具進行靜態分析。對於未提供韌體下載之 IoT 設備，可以嘗試從設備的 UART 接口取得 Debug Shell

對於這次研究的 Tenda AC1206 路由器，Tenda 有在官方網站[1] 公開 AC1206 路由器的韌體。本次研究針對 Tenda AC1206 最新的韌體版本 V15.03.06.23 進行分析與漏洞挖掘

Binwalk 是一個用來分析檔案結構的工具，對研究的韌體版本 V15.03.06.23 分析後發現韌體並未加密，且存在檔案系統。

DECIMAL	HEXADECIMAL	DESCRIPTION
10328	0x2858	LZMA compressed data, properties: 0x5D, dictionary size: 8388608 bytes, uncompressed size: 7070932 bytes
1068530	0x104DF2	MySQL ISAM index file Version 6
2105426	0x202052	Squashfs filesystem, little endian, version 4.0, compression:xz, size: 3700854 bytes, 810 inodes, blocksize: 131072 bytes, created: 2038-04-24 02:46:24

我們可以透過 Binwalk 拆解出檔案系統，分析其中的檔案並列舉攻擊面

```
terry1234@terry1234-virtual-machine:~/Downloads/_US_AC1206V1.0RTL_V15.03.06.23_multi_TD01.bin.extracted$ ls squashfs-root
bin dev etc etc_ro home init lib mnt proc root sbin sys tmp usr var webroot webroot_ro
```

攻擊面選擇

為了方便使用者管理各項路由器的設定，如 VPN、登入帳號密碼等，廠商往往會實作 web 管理介面與 API。本次研究的 Tenda AC1206 路由器也不例外。Web 管理介面由於需要處理 HTTP Request 並與其他設備中的元件互動，在接收 HTTP Request 時常出現 Command Injection, SQL injection, CRLF Injection 等常見網頁漏洞，在儲存與 Parse 資料向其他元件互動時，也常出現 Buffer Overflow 與 Use-After-Free 等漏洞。從上述情況與歷史案例可以知道 Web 管理介面與背後的 Web API 是一個時常出現漏洞的功能，因此我們選擇了 Web 管理介面作為本次研究的攻擊面。

攻擊面分析與韌體模擬

分析檔案系統後發現位於 /bin 底下的 httpd 實作了 Web 管理介面與 Web API

httpd 會先初始化各項設定

```
28 number = 0;
29 init_core_dump();
30 puts("\n\nYes:\n\n      ***** WeLoveLinux***** \n\n Welcome to ...");
31 setup_signals();
32 if ( apmib_init() )
33 {
34     while ( check_network(ipbuf) <= 0 )
35         sleep(1u);
36     sleep(1u);
37     if ( ConnectCfm() )
38     {
39         bopen(0, 61440, 1);
40         memset(value, 0, sizeof(value));
41         if ( !GetValue("lan.webiplansslen", value) )
42             strcpy(value, "0");
43         sslenable = atoi(value);
44         if ( !GetValue("lan.webport", value) )
45             strcpy(value, "80");
46         if ( !GetValue("lan.webipen", webipen) )
47             strcpy(webipen, "0");
48         if ( !strcmp(webipen, "1") )
49         {
50             sslport = atoi(value);
51             port = atoi(value);
52         }
53         v4 = getLanIfName();
54         if ( getIfIp(v4, br0IP) < 0 )
55         {
56             GetValue("lan.ip", value);
57             strcpy(g_lan_ip, value);
58             memset(&lan_ip_info, 0, sizeof(lan_ip_info));
```

包含以下內容

1. 使用 setup_signals 初始化各個 signal 的 handler

```
void setup_signals()
{
    signal(16, (__sig_handler_t)Term_Sig_entry);
    signal(17, (__sig_handler_t)Term_Sig_entry);
    signal(15, (__sig_handler_t)Term_Sig_entry);
    signal(13, (__sig_handler_t)1);
    signal(1, (__sig_handler_t)Bad_Sig_entry);
    signal(18, (__sig_handler_t)1);
    signal(6, (__sig_handler_t)Bad_Sig_entry);
    signal(6, (__sig_handler_t)1);
    signal(14, (__sig_handler_t)Bad_Sig_entry);
    signal(8, (__sig_handler_t)Bad_Sig_entry);
    signal(3, (__sig_handler_t)Bad_Sig_entry);
    signal(13, (__sig_handler_t)1);
    signal(10, (__sig_handler_t)Bad_Sig_entry);
    signal(7, (__sig_handler_t)Bad_Sig_entry);
    signal(22, (__sig_handler_t)Bad_Sig_entry);
    signal(29, (__sig_handler_t)Bad_Sig_entry);
    signal(12, (__sig_handler_t)Bad_Sig_entry);
    signal(5, (__sig_handler_t)Bad_Sig_entry);
    signal(28, (__sig_handler_t)Bad_Sig_entry);
    signal(30, (__sig_handler_t)Bad_Sig_entry);
    signal(31, (__sig_handler_t)Bad_Sig_entry);
}
```

2. 使用 init_core_dump() 開啟 core dump

```
1 int init_core_dump()
2 {
3     int result; // $v0
4     struct rlimit v1; // [sp+20h] [+20h] BYREF
5     char v2[24]; // [sp+28h] [+28h] BYREF
6     char v3[256]; // [sp+40h] [+40h] BYREF
7
8     strcpy(v2, "/tmp/core-%e-%p-%t-%s");
9     memset(v3, 0, sizeof(v3));
10    memset(&v1, 0, sizeof(v1));
11    if ( !getrlimit(4, &v1) )
12        printf("%s %d: rlim_cur = %d, rlim_max = %d\n", "init_core_dump", 1917, v1.rlim_cur, v1.rlim_max);
13    v1.rlim_cur = 5242880;
14    v1.rlim_max = 5242880;
15    if ( setrlimit(4, &v1) )
16    {
17        perror("setrlimit");
18    }
19    else
20    {
21        printf("%s %d: open core dump success\n", "init_core_dump", 1926);
22        sprintf(v3, "%s > /proc/sys/kernel/core_pattern", v2);
23        doSystemCmd("echo %s > /proc/sys/kernel/core_pattern", v2);
24    }
25    memset(&v1, 0, sizeof(v1));
26    result = getrlimit(4, &v1);
27    if ( !result )
28        result = printf("%s %d: rlim_cur = %d, rlim_max = %d\n", "init_core_dump", 1935, v1.rlim_cur, v1.rlim_max);
29    return result;
30 }
```

3. apmib_init() 會先執行 lock，避免多個 process 同時使用變數，接著載入 hw(hardware monitor) config、default system config 和 current system config，如果載入失敗的話就釋放空間並 return 0。載入完後 unlock 並 return 1。

```

9  apmib_sem_lock(); |
10  if ( !pMib )
11  {
12      v0 = apmib_hwconf();
13      if ( !v0 )
14      {
15 LABEL_6:
16          apmib_sem_unlock();
17          return 0;
18      }
19      fbss = v0;
20      v1 = apmib_dsconf();
21      if ( !v1 )
22      {
23          v2 = fbss;
24          v3 = 0;
25 LABEL_5:
26          apmib_shm_free(v2, v3);
27          goto LABEL_6;
28      }
29      pMibDef = v1;
30      v5 = apmib_csconf();
31      if ( !v5 )
32      {
33          apmib_shm_free(fbss, 0);
34          v2 = pMibDef;
35          v3 = 1;
36          goto LABEL_5;
37      }
38      pMib = v5;
39  }
40  apmib_sem_unlock();
41  return 1;
42 }
```

4. check_network() 負責檢查網卡設備是否存在以及網路是否連接，失敗則會等待1秒並重試。

```
1 bool __fastcall check_network(int a1)
2 {
3     int v1; // $v0
4
5     v1 = getLanIfName();
6     return getIfIp(v1, a1) >= 0;
7 }
```

```
1 int getLanIfName()
2 {
3     return get_eth_name(0);
4 }
```

```
1 const char *__fastcall get_eth_name(int a1)
2 {
3     const char *result; // $v0
4
5     switch ( a1 )
6     {
7         case 0:
8             result = "br0";
9             break;
10        case 1:
11            result = "br1";
12            break;
13        case 6:
14            result = "vlan1";
15            break;
16        case 10:
17            result = "eth1";
18            break;
19        case 11:
20            result = "eth2";
21            break;
22        case 12:
23            result = "eth3";
24            break;
```

在確認上述檢查硬體設備與初始化結果皆正常執行後，會初始化並啟動 Web Server

```
● 66     memset(&info, 0, sizeof(info));
● 67     info.ip = inet_addr(g_lan_ip);
● 68     tpi_talk_to_kernel(5, &info, &number, 0, 0, 0);
● 69     getwebuserpwd(1);
● 70     getwebuserpwd(0);
● 71     v5 = getpid();
● 72     doSystemCmd("echo %d > %s", v5, "/etc/httpd.pid");
● 73     if ( initWebs() >= 0 )
74     {
● 75         memset(loginUserInfo, 0, sizeof(loginUserInfo));
● 76         signal(15, (__sighandler_t)websTermSigHandler);
● 77         signal(9, (__sighandler_t)websTermSigHandler);
● 78         signal(14, (__sighandler_t)web_auth_timer);
● 79         alarm(0x3Cu);
● 80         loop_cnt = 0;
● 81         mallopt(-1, 0);
● 82         mallopt(-3, 2048);
● 83         getpid();
● 84         while ( !finished )
85         {
● 86             if ( socketSelect(-1, 1000) > 0 )
● 87                 socketProcess(-1);
● 88             websCgiCleanup();
● 89             emfSchedProcess();
● 90             if ( !(++loop_cnt % 100) )
● 91                 malloc_trim(0);
● 92             if ( g_hw_nat_switch )
93             {
● 94                 g_hw_nat_switch = 0;
● 95                 hw_nat_config(1);
96             }
```

如果上述的初始化以及連接 server 等動作沒有成功的話，就會跳出錯誤訊息並結束程式。

```
110     else
111     {
112         puts("main -> initWebs failed");
113         result = -1;
114     }
115 }
116 else
117 {
118     printf("connect cfm failed!");
119     result = 0;
120 }
121 }
122 else
123 {
124     puts("Initialize AP MIB failed !");
125     result = -1;
126 }
127 return result;
128 }
```

initWebs() 中則會對網頁使用的各項資訊做初始化，包含 Default Page、Web 管理介面的密碼等

設定完成後會嘗試開啟 Web Server 並設定 API route

```
memset(wbuf, 0, sizeof(wbuf));
doSystemCmd("echo 0 > /proc/sys/net/ipv4/tcp_timestamps");
socketOpen();
inet_aton(g_lan_ip, &intaddr);
strcpy(webdir, rootWeb);
websSetDefaultDir(webdir);
cp = inet_ntoa(intaddr);
if ( strlen(cp) + 1 >= 0x80 )
    v0 = 128;
else
    v0 = strlen(cp) + 1;
ascToUni(wbuf, cp, v0);
websSetIpaddr(wbuf);
if ( strlen(host) + 1 >= 0x80 )
    v1 = 128;
else
    v1 = strlen(host) + 1;
ascToUni(wbuf, host, v1);
websSetHost(wbuf);
websSetDefaultPage("main.html");
websSetPassword(password);
if ( websOpenServer(port, retries) >= 0 )
{
    websUrlHandlerDefine(
        byte_506EEC,
        0,
        0,
        (int (*)(webs_t, char_t *, char_t *, int, char_t *, char_t *, char_t *))R7WebsSecurityHandler,
        1);
    websUrlHandlerDefine(
        "/goform",
        0,
        0,
        (int (*)(webs_t, char_t *, char_t *, int, char_t *, char_t *, char_t *))websFormHandler,
        0);
}
```



```

        webUrlHandlerDefine(
            "/cgi-bin",
            0,
            0,
            (int (*)(webs_t, char_t *, char_t *, int, char_t *, char_t *, char_t *))webs_Tenda_CGI_BIN_Handler,
            0);
    webUrlHandlerDefine(
        byte_506EEC,
        0,
        0,
        (int (*)(webs_t, char_t *, char_t *, int, char_t *, char_t *, char_t *))websDefaultHandler,
        2);
    formDefineTenda();
    webUrlHandlerDefine(
        "/",
        0,
        0,
        (int (*)(webs_t, char_t *, char_t *, int, char_t *, char_t *, char_t *))websHomePageHandler,
        0);
    return 0;
}
else
{
    printf("%s %d: websOpenServer failed\n", "initWebs", 520);
    return -1;
}

```

其中 formDefineTenda() 的功能為設定 API route 與對應的 Handler 實作。

這個 Function 包含了大量的 API Handler 與對應的 API route，透過分析此處可以快速了解各個 API 的功能並挖掘漏洞

```

void formDefineTenda()
{
    websAspDefine("TendaGetLongString", (int (*)(int, webs_t, int, char_t **))aspTendaGetLongString);
    websAspDefine("aspTendaGetStatus", (int (*)(int, webs_t, int, char_t **))aspTendaGetStatus);
    websFormDefine("updateUrlLog", (void (*)(webs_t, char_t *, char_t *))updateUrlLog);
    websFormDefine("SysStatusHandle", (void (*)(webs_t, char_t *, char_t *))fromSysStatusHandle);
    websFormDefine("GetWanStatus", (void (*)(webs_t, char_t *, char_t *))formGetWanStatus);
    websFormDefine("GetSysInfo", (void (*)(webs_t, char_t *, char_t *))formGetSysInfo);
    websFormDefine("GetWanStatistic", (void (*)(webs_t, char_t *, char_t *))formGetWanStatistic);
    websFormDefine("GetAllWanInfo", (void (*)(webs_t, char_t *, char_t *))formGetAllWanInfo);
    websFormDefine("GetWanNum", (void (*)(webs_t, char_t *, char_t *))formGetWanNum);
    websAspDefine("aspGetWanNum", (int (*)(int, webs_t, int, char_t **))aspGetWanNum);
    websFormDefine("getPortStatus", (void (*)(webs_t, char_t *, char_t *))formGetPortStatus);
    websFormDefine("GetSystemStatus", (void (*)(webs_t, char_t *, char_t *))formGetSystemStatus);
    websFormDefine("GetRouterStatus", (void (*)(webs_t, char_t *, char_t *))formGetRouterStatus);
    websFormDefine("setNotUpgrade", (void (*)(webs_t, char_t *, char_t *))formsetNotUpgrade);
}

```

部分 API 存在多層的 function 呼叫，為了方便了解參數如何被傳遞與驗證挖掘漏洞時的一些猜想，我們決定使用 qemu 配合 gdb-multiarch 進行模擬

httpd 的指令集架構是 mips

因此使用 qemu-mipsel-static 模擬，先將 qemu-user-static 複製到 squashfs-root 底下。使用以下指令模擬。

```
sudo chroot ./ ./qemu-mipsel-static ./bin/httpd
```

由於沒有硬體設備，在執行 apmib_init() 時會出現錯誤，所以對 httpd 做 patch

接著會出現 connect cfm failed!，所以同樣要對 ConnectCfm() 做 patch。

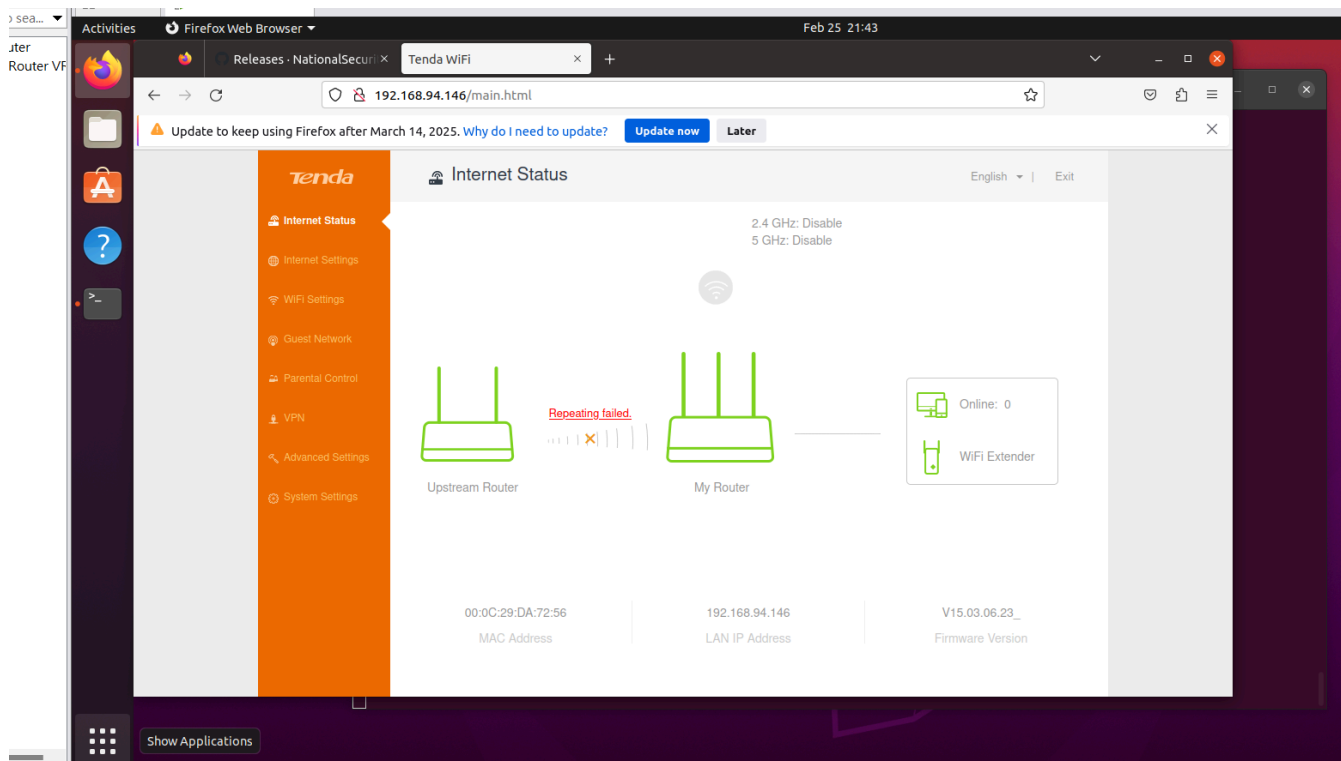
由於在 check_network() 中，會確認系統中是否存在一張名為 br0 的網卡，我們可以新增一張網卡解決此問題。

```
sudo apt install uml-utilities bridge-utils
sudo brctl addbr br0
sudo brctl addif br0 ens33
sudo ifconfig br0 up
sudo dhclient br0
```

最後將 patch 過的 httpd_patched 匯出，放入 ./squashfs-root/bin，使用以下指令嘗試啟動。

```
cd squashfs-root
sudo chroot ./ ./qemu-mipsel-static ./bin/httpd_patched
```

成功模擬此 Web 管理介面與對應的 Web API



漏洞介紹

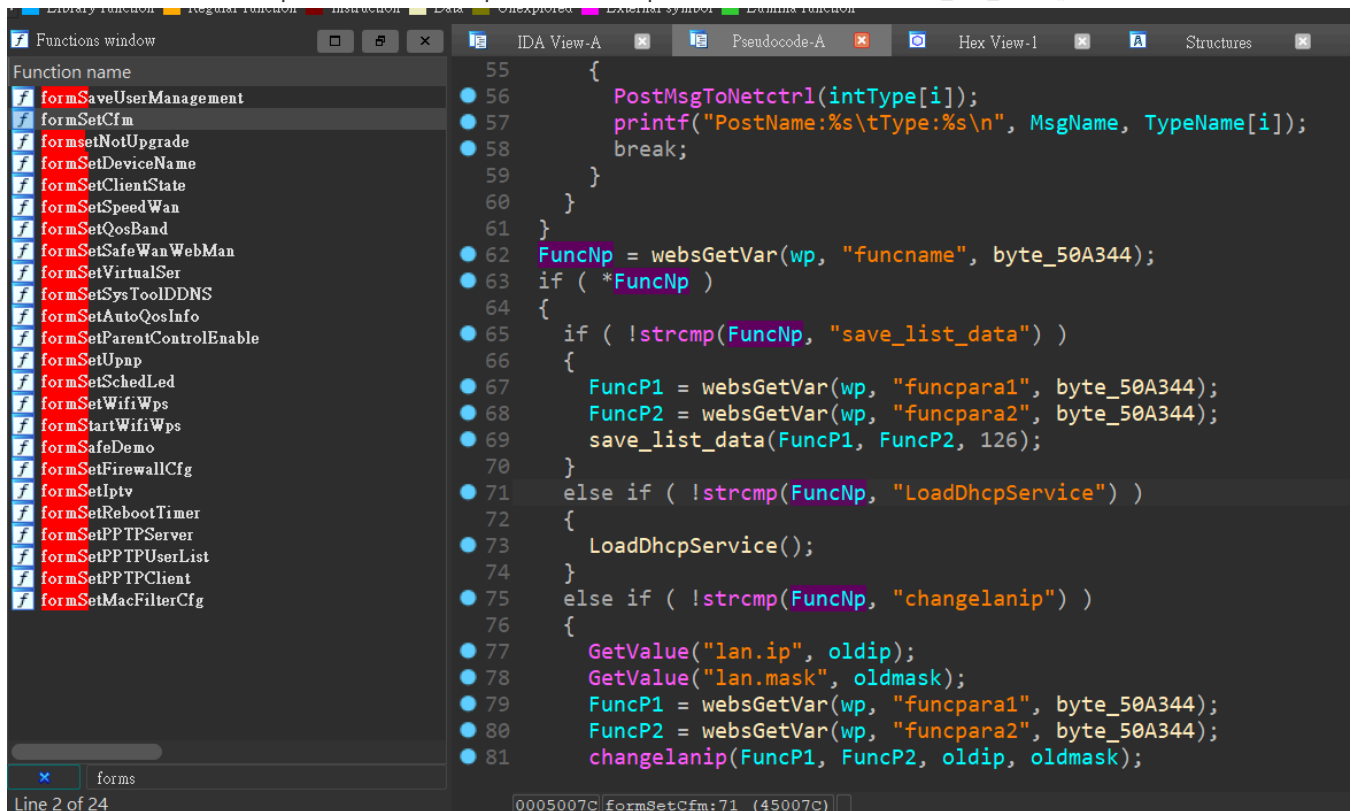
針對 formDefineTendDa() 中的 184 個 API 做分析後，我們發現了數個漏洞，其中有部分漏洞為在 Tenda AC1206 已經被 Assigned 的 CVE

以下內容主要介紹我們發現並嘗試回報的兩個漏洞

韌體與 PoC 可在下方的連接中找到：<https://github.com/qingwei4/Router-VR>

CVE-2024-53621

在 formSetCfm() 這個函式中，會透過 websGetVar() 從 HTTP Request 取得參數值。其中，圖片中的程式碼片段會從 HTTP request 中獲取 funcname 參數並將其儲存於指向 Bss 區段的 FuncNp 指標中。如果此參數與 save_list_data 相同，則會從 HTTP Request 中獲取 funcpara1 與 funcpara2 兩個參數傳入 save_list_data() 中



```
55 {
56     PostMsgToNetctrl(intType[i]);
57     printf("PostName:%s\tType:%s\n", MsgName, TypeName[i]);
58     break;
59 }
60 }
61 }
62 FuncNp = websGetVar(wp, "funcname", byte_50A344);
63 if ( *FuncNp )
64 {
65     if ( !strcmp(FuncNp, "save_list_data") )
66     {
67         FuncP1 = websGetVar(wp, "funcpara1", byte_50A344);
68         FuncP2 = websGetVar(wp, "funcpara2", byte_50A344);
69         save_list_data(FuncP1, FuncP2, 126);
70     }
71     else if ( !strcmp(FuncNp, "LoadDhcpService") )
72     {
73         LoadDhcpService();
74     }
75     else if ( !strcmp(FuncNp, "changelanip") )
76     {
77         GetValue("lan.ip", oldip);
78         GetValue("lan.mask", oldmask);
79         FuncP1 = websGetVar(wp, "funcpara1", byte_50A344);
80         FuncP2 = websGetVar(wp, "funcpara2", byte_50A344);
81         changelanip(FuncP1, FuncP2, oldip, oldmask);
82     }
83 }
```

save_list_data() 中的 sprintf() 執行前並沒有先檢查 FuncP1 的長度，這會讓使用者有機會透過輸入超過 mib_name

buffer 大小的資料導致 stack buffer overflow。

Unexplored External symbol Lumina function

```
IDA View-A Pseudocode-A Hex View-1
1 void __cdecl save_list_data(char *funcP1, char *funcP2, char c)
2 {
3     char *i; // $v0
4     int count; // [sp+18h] [+18h]
5     int counta; // [sp+18h] [+18h]
6     int countb; // [sp+18h] [+18h]
7     char *q; // [sp+1Ch] [+1Ch]
8     char *p; // [sp+20h] [+20h]
9     char mib_name[64]; // [sp+24h] [+24h] BYREF
10    char mib_value[256]; // [sp+64h] [+64h] BYREF
11    char ct[8]; // [sp+164h] [+164h] BYREF
12
13    memset(mib_name, 0, sizeof(mib_name));
14    memset(mib_value, 0, sizeof(mib_value));
15    if ( strlen(funcP2) >= 5 )
16    {
17        counta = 1;
18        p = funcP2;
19        for ( i = strchr(funcP2, c); ; i = strchr(q + 1, c) )
20        {
21            q = i;
22            if ( !i )
23                break;
24            *i = 0;
25            memset(mib_name, 0, sizeof(mib_name));
26            sprintf(mib_name, "%s.list%d", funcP1, counta);
27            SetValue(mib_name, p);
28            p = q + 1;
29            ++counta;
30        }
31        memset(mib_name, 0, sizeof(mib_name));
32        sprintf(mib_name, "%s.list%d", funcP1, counta);
```

CVE-2025-50814

當呼叫 fromAdvSetMacMtuWan() 時，會執行 check_param_changed() 這個函式。

```
1 void __cdecl fromAdvSetMacMtuWan(webs_t wp, char_t *path, char_t *query)
2 {
3     int *v3; // $v0
4     unsigned int j; // $v1
5     int wan_id; // [sp+70h] [+70h] BYREF
6     int i; // [sp+74h] [+74h]
7     int wann; // [sp+78h] [+78h]
8     int error_code; // [sp+7Ch] [+7Ch]
9     char mib_value[16]; // [sp+80h] [+80h] BYREF
10    char ret_buf[64]; // [sp+90h] [+90h] BYREF
11    WAN_ARGUMENT wan_set_param[2]; // [sp+D0h] [+D0h] BYREF
12    WAN_ARGUMENT v13; // 0:$a2.8,8:^10.88
13
14    error_code = 0;
15    wan_id = 0;
16    memset(mib_value, 0, sizeof(mib_value));
17    memset(ret_buf, 0, sizeof(ret_buf));
18    memset(wan_set_param, 0, sizeof(wan_set_param));
19    GetValue("wans.flag", mib_value);
20    wann = atoi(mib_value);
21    for ( i = 0; i < wann; ++i )
22    {
23        wan_id = i + 1;
24        if ( check_param_changed(wp, i + 1, &wan_set_param[i]) )// bof
25        {
26            v3 = &wan_id + 24 * i;
27            for ( j = 0; j < 0x58; ++j )
28                v13.wan_mtu[j - 8] = *((_BYTE *)v3 + j + 104);
29            *(_QWORD *)v13.wan_connecttype = *((_QWORD *)v3 + 12);
30            error_code = formWanArgumentSet(wp, wan_id, v13);
31        }
32    }
```

在 `check_param_changed()` 中，會使用 `websGetVar()` 來從 HTTP Request 的參數中分別獲取 `wanMTU`, `wanSpeed` 與 `CloneType` 的值，

並分別儲存於 Bss 區段的 `Var`, `v5`, `v6` 中，隨後會直接使用 `strcpy` 將其複製到 `stack` 上，並未經過任何檢查，導致 `stack buffer overflow`

```
int __cdecl check_param_changed(webs_t wp, int wan_id, WAN_ARGUMENT *wan_param)
{
    char_t *Var; // $v0
    char_t *v5; // $v0
    char_t *v6; // $v0
    char_t *v7; // $v0
    char_t *v8; // $v0
    char_t *v9; // $v0
    char_t *v10; // $v0
    int wan_connect_type; // [sp+18h] [+18h]
    int change_flag; // [sp+1Ch] [+1Ch]
    char mib_name[32]; // [sp+20h] [+20h] BYREF
    char mib_value[16]; // [sp+40h] [+40h] BYREF
    char clone_type[20]; // [sp+50h] [+50h] BYREF

    memset(mib_name, 0, sizeof(mib_name));
    memset(mib_value, 0, sizeof(mib_value));
    memset(clone_type, 0, 16);
    change_flag = 0;
    sprintf(mib_name, "wan%d.connecttype", wan_id);
    GetValue(mib_name, wan_param);
    wan_connect_type = atoi((const char *)wan_param);
    if ( wan_param )
    {
        if ( wan_id == 1 )
        {
            Var = websGetVar(wp, "wanMTU", byte_50F88C);
            strcpy(wan_param->wan_mtu, Var);
            v5 = websGetVar(wp, "wanSpeed", "0");
            strcpy(wan_param->wan_speed, v5);
            v6 = websGetVar(wp, "cloneType", "0");
            strcpy(wan_param->clone_type, v6);
        }
    }
}
```

回報過程

嘗試聯繫廠商後並未獲得回覆，因此我們嘗試透過 mitre 提供的 CVE form 回報

最終獲得 CVE-2024-53621 與 CVE-2025-50813 兩個 CVE 編號

CVE-2024-53621

```
> [Vulnerability Type]
> Buffer Overflow
>
> -----
>
> [Vendor of Product]
> Tenda
>
> -----
>
> [Affected Product Code Base]
> AC1206 1200M 11ac - US_AC1206V1.0RTL_V15.03.06.23_multi_TD01
>
> -----
>
> [Affected Component]
> function formSetCfm()
>
> -----
>
> [Attack Type]
> Remote
>
> -----
>
> [Impact Code execution]
> true
>
> -----
>
> [Attack Vectors]
> user can send POST request to http://{ip}:{port}/goform/setcfm with a well-crafted payload to achieve RCE or DoS.
>
> -----
>
> [Reference]
> https://drive.google.com/file/d/1b7DlkG7XVmJmCxDrX7u1X7CAk-IOviBX/view?usp=sharing
```

Use CVE-2024-53621.

```
> [Suggested description]
> AC1206 1200M 11ac firmware version
> US_AC1206V1.0RTL_V15.03.06.23_multi_TD01 has buffer overflow
> vulnerability in formSetCfm()
```

CVE-2025-50814

```
> [Affected Product Code Base]
> AC1206 1200M 11ac - US_AC1206V1.0RTL_V15.03.06.23_multi_TD01
>
> -----
>
> [Affected Component]
> function fromAdvSetMacMtuWan()
>
> -----
>
> [Attack Type]
> Remote
>
> -----
>
> [Impact Code execution]
> true
>
> -----
>
> [Impact Denial of Service]
> true
>
> -----
>
> [Attack Vectors]
> Attacker can send POST request with well-crafted payload to /goform/AdvSetMacMtuWan to achive DoS or RCE
>
> -----
>
> [Discoverer]
> Ching-Wei, Huang & Chun-Yu, Lin
>
> -----
>
> [Reference]
> http://ac1206.com
> http://tenda.com
> https://drive.google.com/file/d/1p5YOW0zhdHB4z0sB8wBbCZ2\_swvb8ZID/view?usp=sharing
```

Use CVE-2025-50814.

Reference

1. 韌體下載：<https://www.tenda.com.cn/product/help/AC1206#download>
2. 漏洞細節與 PoC：<https://github.com/qingwei4/Router-VR>
3. CVE form：<https://cveform.mitre.org/>