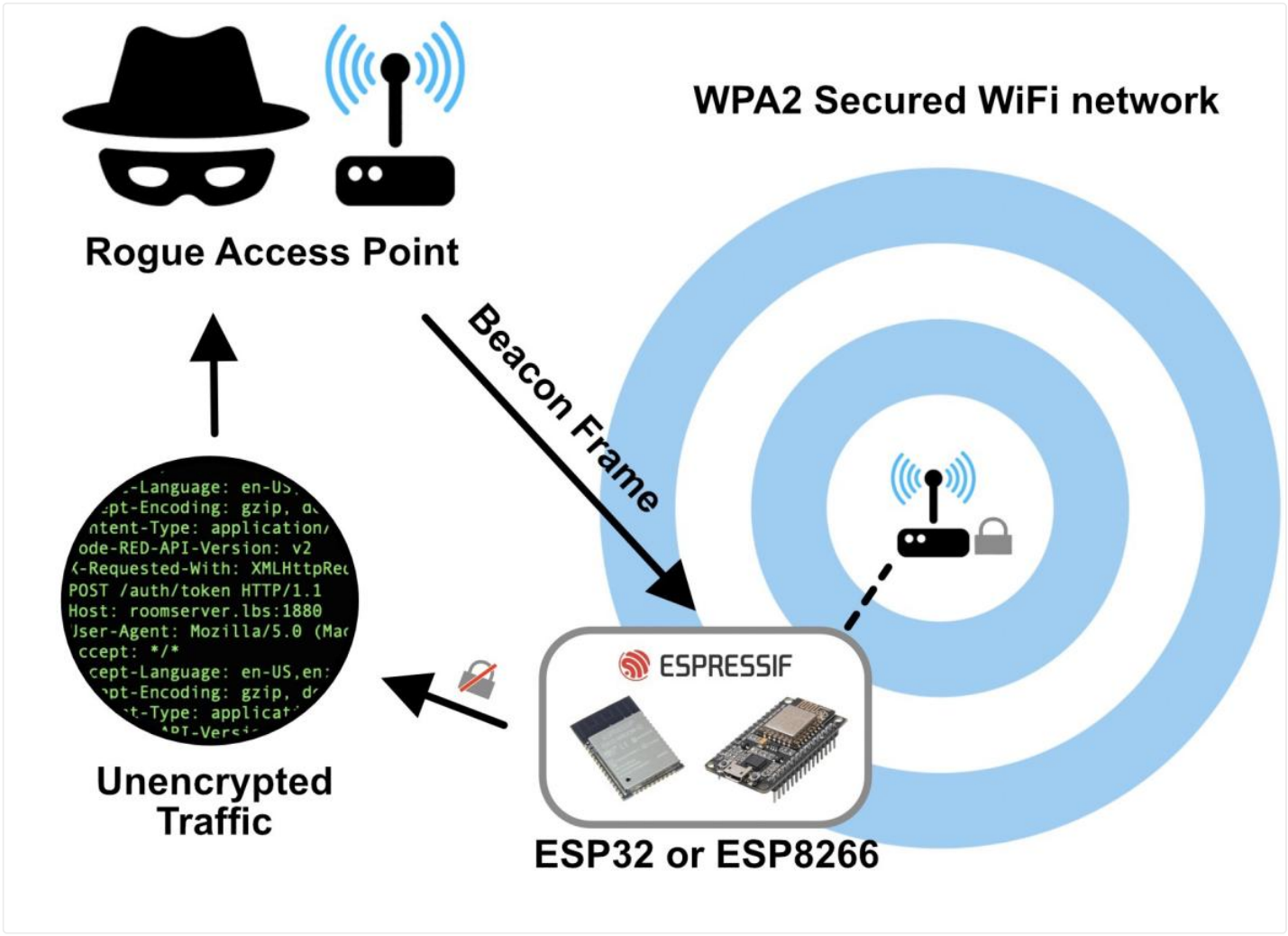
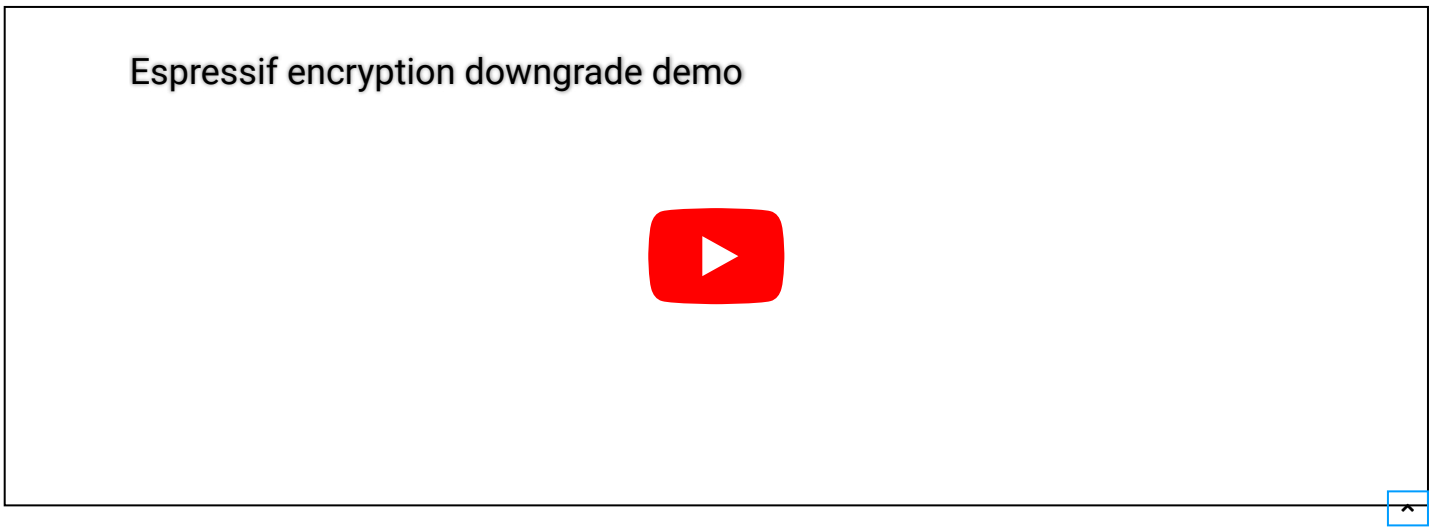


WPA2 Authenticationmode downgrade in Espressif microprocessors (CVE-2020-12638)

23 July 2020

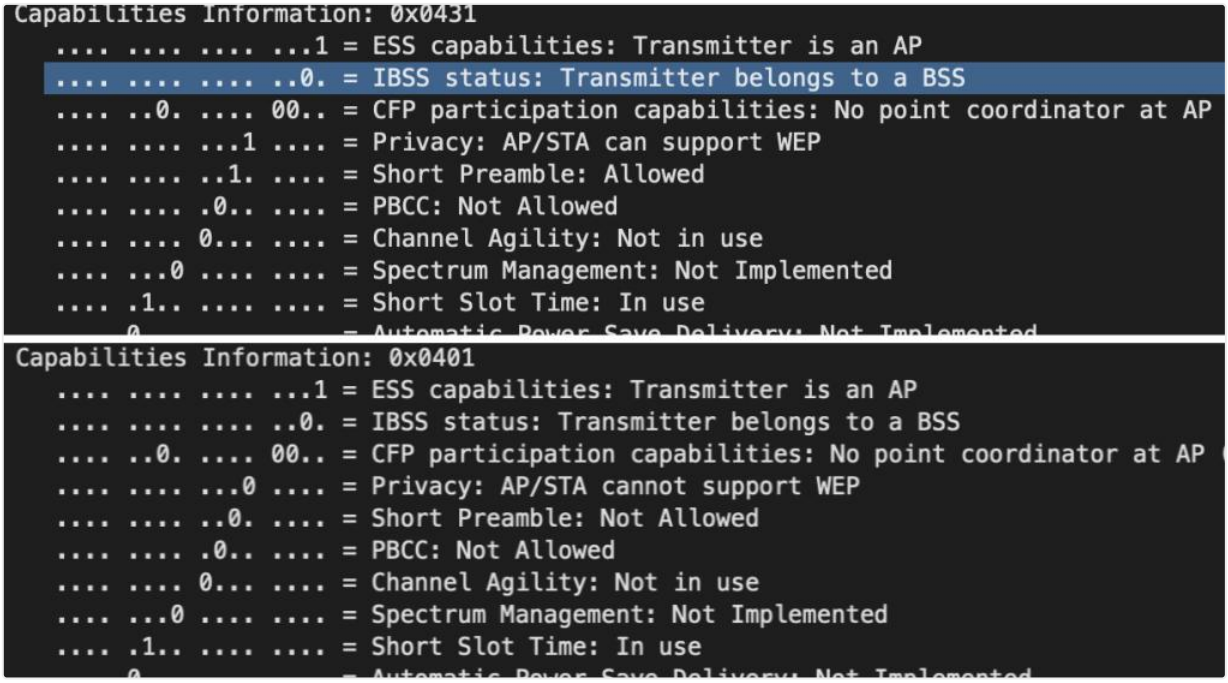
Update 26.2.21: Added Info about Arduino ESP32 patches as of 1.0.5 release

Update 26.7.20: Added note about esp8266 arduino core implementing workaround While working on my master thesis about "Security of open-source home automation software" I discovered a serious security issue in the **Espressif Systems (https://www.espressif.com)** microprocessor SDKs. Espressif has been working on the patch of this vulnerability that has been **deployed to several versions** of their SDKs. Additionally a simple **workaround** can be applied in the code if upgrading to a patched version is no immediate option.



Description

This vulnerability allows forcing the ESP8622 and ESP32 chip families into downgrade their WiFi authentication mode, effectively disabling their encryption entirely. Using a **channel switch attack** an adversary can easily gain a man-in-the-middle position and **read**, **replay** and **manipulate** any unprotected traffic of the device. It works by sending a beacon frame with the same data as the WiFi network that the ESP is currently connected to, but switching the **Privacy** bit in the authentication header to 0. This will cause the Espressif device to switch to the **OPEN** authentication mode and send out **unencrypted traffic** until it receives a beacon frame from the original access point again. To stabilize the attack a so called Channel Switch Announcement can be sent to force the ESP chip to switch to a different wireless channel. This way it will not receive the original access point beacons anymore and keep sending unencrypted communication to the rouge access point. Here are two Wire-shark screenshots comparing the original (upper image) and the forged (lower image) 802.11 beacon frames:



Why is this an issue?

ESP8266 and ESP32 are commonly used in many different applications from hobby to commercial applications. When the application code running on the chip does not ensure authentication, encryption and verification of it's communication as it is the case in many open-source home automation systems such as Tasmota or ESPHome[1] it is possible to **manipulate** these devices without having the WiFi credentials in the first place.

Current state of the patch

- Espressif has patched the issue in the following SDKs: *ESP-IDF* [2]

- ESP8266 NONOS SDK (master version) [3]

- ESP8266 RTOS SDK [4]

- ESP32 Arduino core since 1.0.5-rc1 or this commit (https://github.com/espressif/arduino-esp32/commit/22b427df) [5]

- ESP8266 Arduino core [6] (This one is not directly maintained by espressif, woraround is in place (https://github.com/esp8266/Arduino/pull/7486)

Workaround

By increasing the log level of the Espressif SDKs it is easy to detect the attack. The WiFi stack reports it's **WIFI_EVENT_STA_AUTHMODE_CHANGED** event[7] , including the information of the old and the new mode used. This makes it trivial for the application code on the chip to enforce a disconnect once a switch from a more secure to a less secure authentication mode is detected, effectively mitigating the risk of a compromise of information. A simple proof-of-concept mitigation for the ESP32 could look like this: https://github.com/s00500/esphome/commit/52f11b8a386f5ba7ed2904a4abc4ee883c5b45d3 (https://github.com/s00500/esphome/commit/52f11b8a386f5ba7ed2904a4abc4ee883c5b45d3)

(In NONOS SDK: **EVENT_STA_MODE_AUTHMODE_CHANGE**, in RTOS SDK **SYSTEM_EVENT_STA_AUTHMODE_CHANGE**)

In the patched version of the ESP-IDF the event has been removed.

Further Links

See Espressif's Security Advisory published here: https://www.espressif.com/sites/default/files/advisory (https://www.espressif.com/sites/default/files/advisorydownloads/Security%20advisory%20authentication%20bypass.pdf)*downloads/Security%20advisory (https://www.espressif.com/sites/default/files/advisorydownloads/Security%20advisory%20authentication%20bypass.pdf)*%20authentication%20bypass.pdf (https://www.espressif.com/sites/default/files/advisorydownloads/Security%20advisory%20authentication%20bypass.pdf)

Proof of Concept implementation: **Will be published later this year**

CVE-2020-12638 Entry in the CVE Database: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-12638 (https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-12638)

If this was helpful or you have any questions consider writing me an email (mailto:lukas@lbsfilm.at) or buying me a coffee (https://paypal.me/lukasbachschwell) ☕

1. in their default configurations ↩
2. (https://github.com/espressif/esp-idf/commit/179292f9b3fe8fdbcccf0a9d2c0f50d394fddc10)https://github.com/espressif/esp-idf/commit/179292f9b3fe8fdbcccf0a9d2c0f50d394fddc10 (https://github.com/espressif/esp-idf/commit/179292f9b3fe8fdbcccf0a9d2c0f50d394fddc10) ↩
3. (https://github.com/espressif/ESP8266)https://github.com/espressif/ESP8266 (https://github.com/espressif/ESP8266)*NONOS*SDK ↩
4. (https://github.com/espressif/ESP8266)https://github.com/espressif/ESP8266 (https://github.com/espressif/ESP8266)*RTOS*SDK ↩
5. (https://github.com/espressif/arduino-esp32)https://github.com/espressif/arduino-esp32 (https://github.com/espressif/arduino-esp32) ↩
6. (https://github.com/esp8266/Arduino)https://github.com/esp8266/Arduino (https://github.com/esp8266/Arduino) ↩
7. (https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/wifi.html#wifi-event-sta-authmode-change)https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/wifi.html#wifi-event-sta-authmode-change (https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/wifi.html#wifi-event-sta-authmode-change) ↩