FragAttacks: summary of findings

Mathy Vanhoef

Draft version 1, 8 March 2021

NEW YORK UNIVERSITY

Table of contents

Design flaws:

Table of contents

- Implementation flaws allowing trivial plaintext injection:

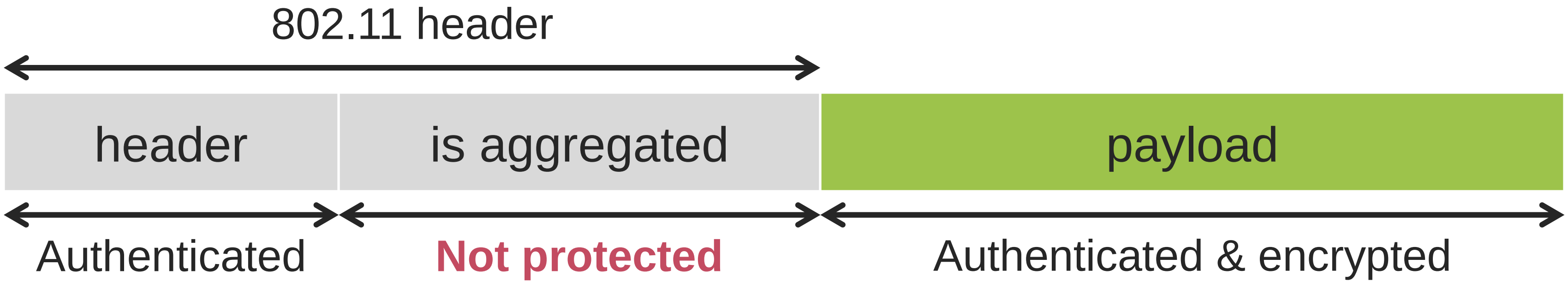
- Other implementation flaws:

Aggregation Attack

CVE-2020-24588

Root cause

- The “is aggregated” flag in the Wi-Fi header is not protected:



- An adversary can flip the “is aggregated” flag
- Payload will be parsed differently → allows packet injection

Impact

Target	Preconditions	Impact
Client	Client connects to attacker's server	Inject packets to client
	AP is vulnerable to CVE-2020-26139	Inject packets to client
AP	Client connects to attacker's server *and* this client uses predictable IP IDs	Inject packets to AP

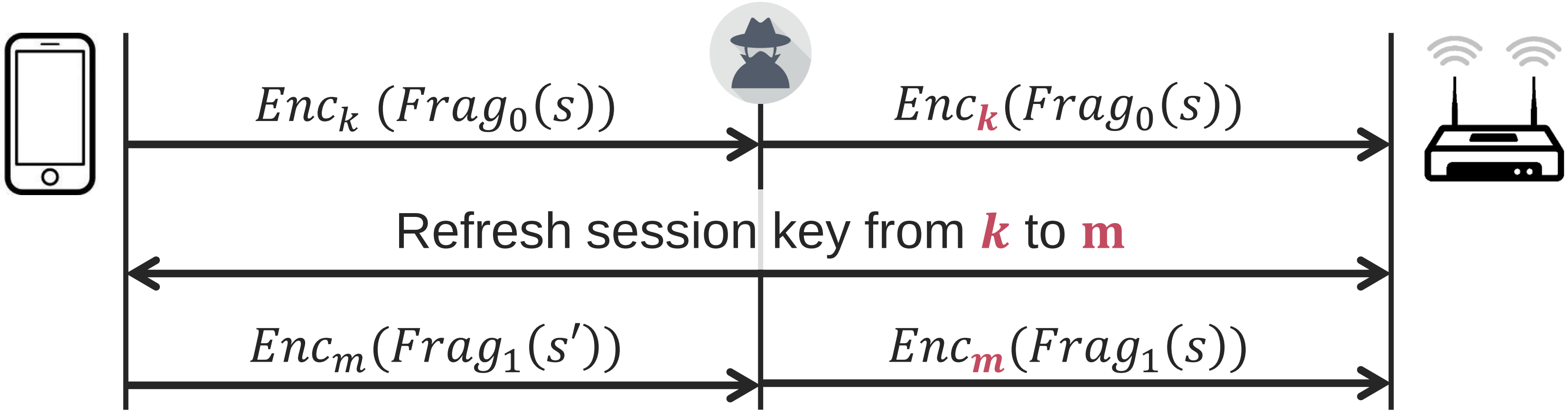
Example attack: make client use a **malicious DNS server** or **bypass the AP's NAT** to directly access local devices

Mixed Key Attack

CVE-2020-24587

Root cause

- Fragments encrypted under different keys are reassembled:



- Receiver will decrypt & reassemble fragments $Frag_0$ and $Frag_1$
- Can be abused to forge frames by mixing fragments

Impact

Target	Preconditions	Impact
AP	Client connects to attacker's server *and* client sens fragmented frames *and* the network refreshes session keys (= unlikely in practice)	Exfiltrate data sent by client
Client	Only a theoretic concern (see paper)	Theoretic (see paper)

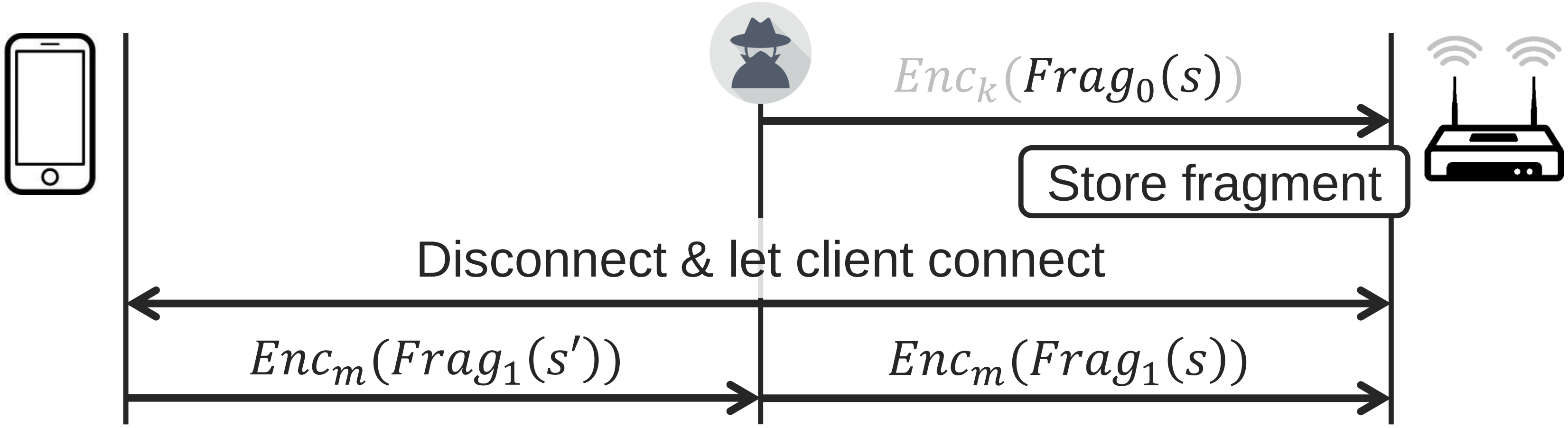
Example attack: **exfiltrate** a web brower **cookie** of the client when plaintext HTTP is used

Fragment Cache Attack

CVE-2020-24586

Root cause

› The fragment cache isn't cleared when (re)connecting:



› Attacker's fragment $Frag_0$ & the client's $Frag_1$ is reassembled

› Can be abused to exfiltrate & forge frames by mixing fragments

Impact

Target	Preconditions	Impact
AP	Target is an enterprise network *and* client sends fragmented frames (fairly unlikely)	Exfiltrate data sent by client *and* inject packets to AP
Client	Client will connect to the adversary's network (but won't trust it) *and* the AP sends fragmented frames (seems unlikely)	Inject packets to client

Example attacks: exfiltrate a plaintext brower cookie, make client use a malicious DNS server, bypass the AP's NAT

Implementation Flaws: trivial plaintext injection

Accepted plaintext frames (CVE-2020-26140 / 26143)

Accepting **plaintext frames** (CVE-2020-26140)

› Examples: some routers, some dongles on Linux/Windows

Accepting **fragmented plaintext frames** (CVE-2020-26143)

› Examples: many dongles on Windows, some FreeBSD APs

→ Can **inject frames** indepedent of network config

Plaintext broadcast fragments (CVE-2020-26145)

Some devices accept **plaintext broadcast fragments**

- Sometimes only accepted while connecting
- Treated as full frames!
- Examples: MacOS, iOS, and Free/NetBSD APs

→ Can **inject frames** indepedent of network config

Cloacked aggregated frames (CVE-2020-26144)

Some accept **aggregate frames that resemble EAPOL** frames

- Sometimes only accepted while connecting
- 2nd subframe of aggregate frame can contain arbitrary data
- Examples: Huawei Y6', Nexus 5X, FreeBSD, LANCOM APs

→ Can **inject frames** indepedent of network config

Implementation flaws with other impact

Non-consective packet numbers (CVE-2020-26146)

Accepting fragments with non-consecutive packet numbers

› Related fragments must have consecutive packet numbers

› But almost **nobody checks this!** Only Linux does.

Can abuse this to **exfiltrate data** sent by a client if:

› The client is tricked into visiting the attacker’s server

› The client sends fragmented frames

Mixed plain/encrypted fragments (CVE-2020-26147)

Some reassemble **mixed plaintext and encrypted fragments**

- Practically all devices are affected

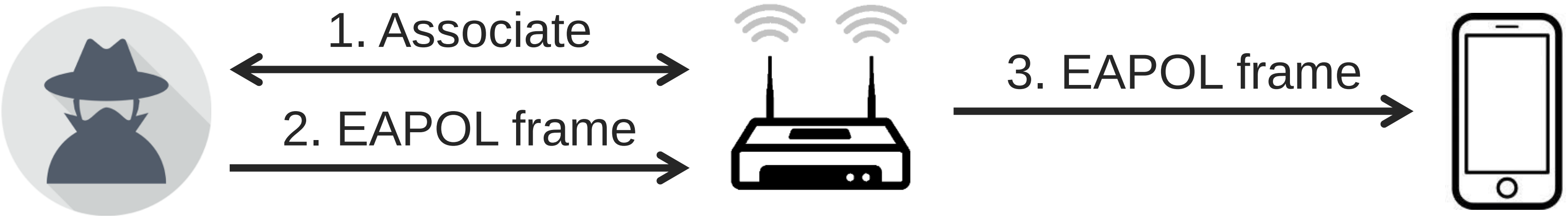
Can **abuse to inject frames**

- If 1st fragment must be encrypted:
  - Inject frames when combined with other vulnerabilities (non-trivial)
- If last fragment must be encrypted:
  - Inject frames when another device sends fragmented frames

Pre-auth EAPOL forwarding (CVE-2020-26139)

Some APs forwards EAPOL frames before sender is authenticated

- Examples: Net/FreeBSD APs and $^2/_4$ home routers



→ Abuse to **inject frames** in combination with aggregation attack (CVE-2020-24588)

No fragmentation support (CVE-2020-26142)

Some devices don't support fragmentation

› They **treat fragmented frames as full frames**

› Examples: OpenBSD and ESP12-F

Abuse to **inject frames** when:

› Another device sends fragmented frames

› This other device visits the attacker's server

Discussion

Practicality vs. impact

Perhaps we're lucky:

- Widespread flaws → relatively trickly to exploit in practice
- Trivial to exploit flaws → not widespread in practice (?)

Important concerns remain:

- Significant #devices affected by trivial to exploit flaws
- **Every Wi-Fi device affected** by one or more flaws
- Combining flaws increases practicality of certain attacks

→ **Patch now** before attack improve!