Samsung Android S10 Association to WPA2-Enterprise (802.1X/EAP) enabled WLAN with Mist AP.
By Gjermund Raaen

During WLPC_EU in Prague 2022 I attended Peter Mackenzie's deep-dive "AP Packet Capture".
During that I discovered a strange association behaviour for my Samsung S10 cell phone to an Mist AP.
Peter and I considered if it could be some kind of FILS authentication, but later testing rejects this.

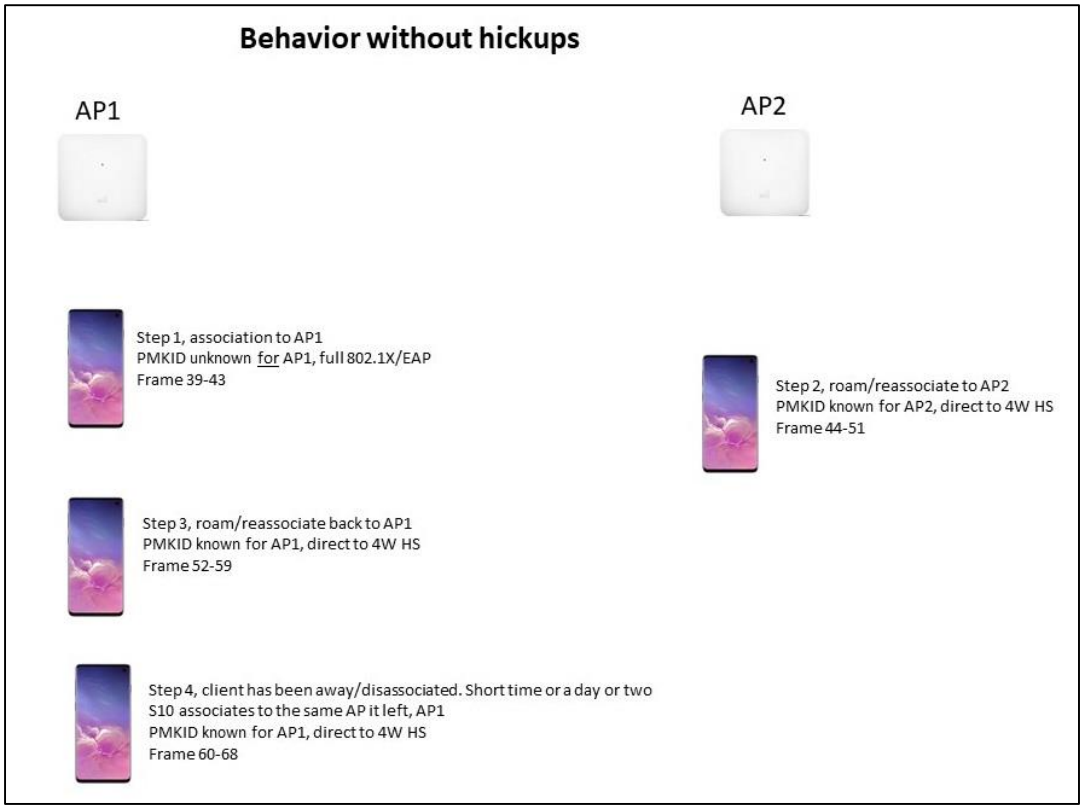
Setup
WLAN on Mist AP is configured with WPA2-Enterprise (802.1X/EAP) authentication and Fast Roaming is set to Default (no roaming). And a configured Radius Authentication Server (JumpCloud).

Default behaviour during association and re-association for WiFi clients on a WPA2-Enterprise WLAN without fast roaming features is to do a full 802.1X/EAP authentication each time it associate and roams (reassociate). During testing with different types of clients these always do full 802.1X/EAP authentication, both during association and re-association (roaming). I have tested several types of clients.

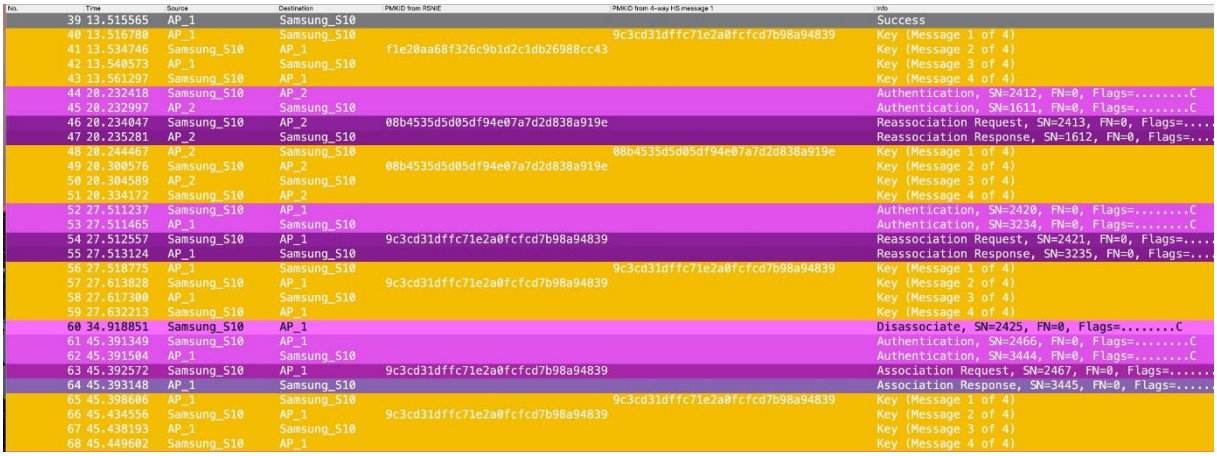
But one does it otherwise, Samsung S10 with Android version 12.

Testing
I replicated the network at work and done several types of operations on a network with two APs, called AP1 and AP2s. The frames are captured and shown with Wireshark, and the use of name in the source and destination columns. The client is called Samsung_S10.
There are mainly two types of behaviour, one without hickups and one with hickups.

Result 1, Seamless associations without hickups



Picture 1. Visualization for steps 1 to 4.



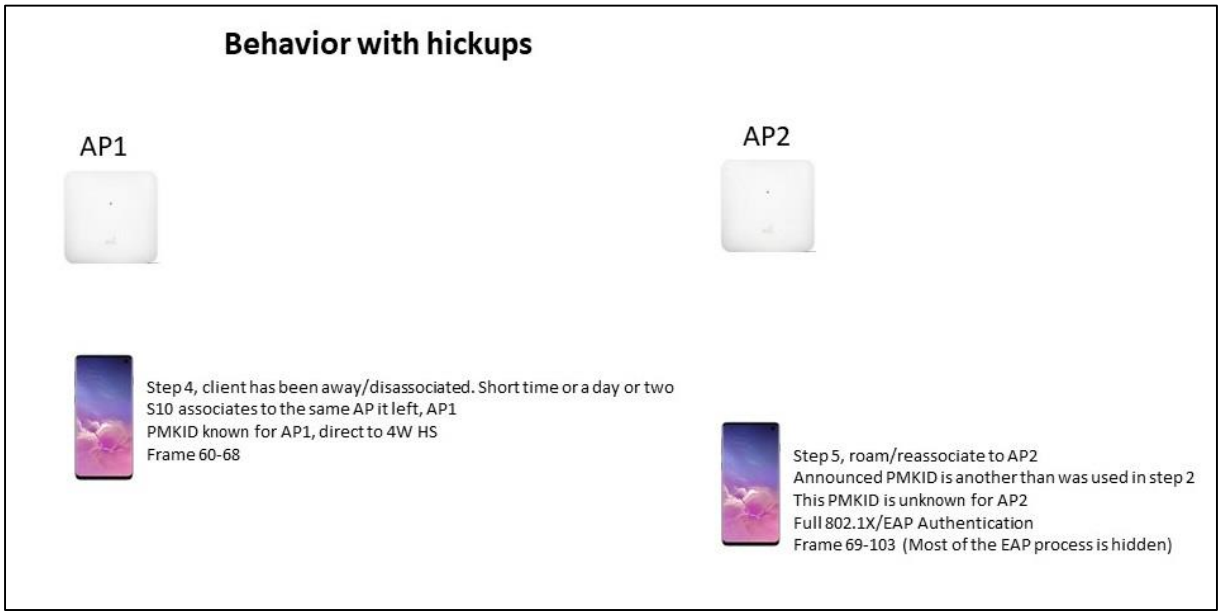
No.	Time	Source	Destination	PMKID from RSNIE	PMKID from 4-way HS message 1	Info
39	13.515565	AP_1	Samsung_S10			Success
40	13.516780	AP_1	Samsung_S10		9c3cd31dffc71e2a0fcfcd7b98a94839	Key (Message 1 of 4)
41	13.534746	Samsung_S10	AP_1	f1e20aa68f326c9b1d2c1db26988cc43		Key (Message 2 of 4)
42	13.540573	AP_1	Samsung_S10			Key (Message 3 of 4)
43	13.561297	Samsung_S10	AP_1			Key (Message 4 of 4)
44	20.232418	Samsung_S10	AP_2			Authentication, SN=2412, FN=0, Flags=........C
45	20.232997	AP_2	Samsung_S10			Authentication, SN=1611, FN=0, Flags=........C
46	20.234047	Samsung_S10	AP_2	08b4535d5d05df94e07a7d2d838a919e		Reassociation Request, SN=2413, FN=0, Flags=....
47	20.235281	AP_2	Samsung_S10			Reassociation Response, SN=1612, FN=0, Flags=...
48	20.244467	AP_2	Samsung_S10		08b4535d5d05df94e07a7d2d838a919e	Key (Message 1 of 4)
49	20.300576	Samsung_S10	AP_2	08b4535d5d05df94e07a7d2d838a919e		Key (Message 2 of 4)
50	20.304589	AP_2	Samsung_S10			Key (Message 3 of 4)
51	20.334172	Samsung_S10	AP_2			Key (Message 4 of 4)
52	27.511237	Samsung_S10	AP_1			Authentication, SN=2420, FN=0, Flags=........C
53	27.511465	AP_1	Samsung_S10			Authentication, SN=3234, FN=0, Flags=........C
54	27.512557	Samsung_S10	AP_1	9c3cd31dffc71e2a0fcfcd7b98a94839		Reassociation Request, SN=2421, FN=0, Flags=....
55	27.513124	AP_1	Samsung_S10			Reassociation Response, SN=3235, FN=0, Flags=...
56	27.518775	AP_1	Samsung_S10		9c3cd31dffc71e2a0fcfcd7b98a94839	Key (Message 1 of 4)
57	27.613828	Samsung_S10	AP_1	9c3cd31dffc71e2a0fcfcd7b98a94839		Key (Message 2 of 4)
58	27.617300	AP_1	Samsung_S10			Key (Message 3 of 4)
59	27.632213	Samsung_S10	AP_1			Key (Message 4 of 4)
60	34.918851	Samsung_S10	AP_1			Disassociate, SN=2425, FN=0, Flags=........C
61	45.391349	Samsung_S10	AP_1			Authentication, SN=2466, FN=0, Flags=........C
62	45.391504	AP_1	Samsung_S10			Authentication, SN=3444, FN=0, Flags=........C
63	45.392572	Samsung_S10	AP_1	9c3cd31dffc71e2a0fcfcd7b98a94839		Association Request, SN=2467, FN=0, Flags=.....
64	45.393148	AP_1	Samsung_S10			Association Response, SN=3445, FN=0, Flags=.....
65	45.398606	AP_1	Samsung_S10		9c3cd31dffc71e2a0fcfcd7b98a94839	Key (Message 1 of 4)
66	45.434556	Samsung_S10	AP_1	9c3cd31dffc71e2a0fcfcd7b98a94839		Key (Message 2 of 4)
67	45.438193	AP_1	Samsung_S10			Key (Message 3 of 4)
68	45.449602	Samsung_S10	AP_1			Key (Message 4 of 4)

Picture 2. Packet capture for step 1 to 4.

Explanation

- Step 1
    - Client associates to AP1 while it announces an earlier used PMKID (not seen in the capture), but AP1 don't know the PMKID.
    - Full 802.1X/EAP-authentication is done, ending with the success frame 39.
    - 4-way handshake.
        - Since AP1 knows the new derived PMK from this EAP-process, authenticator MAC and supplicant MAC it makes a new PMKID, shown in message 1, frame 40.
        - The client responds with the previous and invalid PMKID from the not shown association frame in message 2, in the RSNIE element. This is a security feature according to the 802.11-2020 standard. The new PMKID for this association is the one in message 1.
- Step 2
    - Client roam/reassociates to AP2. It announces an PMKID from an earlier association to AP2 (not shown). AP2 has this PMKID cached and it goes directly to 4-way handshake.
      In this case the PMKID is equal in both the reassociation request, message 1, and 2 frames.
- Step 3
    - Client roams back to AP1. It announces the PMKID from the previous association to AP1, step 1. AP1 has this PMKID cashed and it goes directly to 4-way handshake.
    - In this case the PMKID is equal in both the reassociation request, message 1, and 2 frames.
- Step 4.
    - Client has been disassociated from the network for a period of time. It could be a short time or a day or two
    - Since the client associates (not re-associates) to the same AP (AP1) as it left, the PMKID is known by the client and AP1, and no 802.1X/EAP.
    - This happens when the client has been away for some seconds or a day. Also if the client has been associated to other network or been turned off in between
- Remarks
    - Step 2 and 3, where the 802.1A/EAP process is skipped, is what we know as PMK Caching
    - Step 4, where the client associate (not re-associate) to the same AP as it left and skipped 802.1X/EAP, is also PMK caching process.
    - We are trained to believe PMK caching is only used during re-association or fast roam back
    - But according to 802.11-2020 standards, 12.6.10.3, "Cached PMKSAs and RSNA key management" this method can also be used during association for the lifetime of a PMKSA.
    - From my limited testing, it is only the Samsung S10 that use this feature. It is the same behaviour on a Cisco WLAN with the same setup.

Result 2, Seamless associations with hickups.
Under one condition there is a odd authentication behaviour.



Picture 3. Visualization for steps 4 to 5.

No.	Time	Source	Destination	PMKID from RSNIE	PMKID from 4-way HS massage 1	Info
60	34.918851	Samsung_S10	AP_1			Disassociate, SN=2425, FN=0, Flags=
61	45.391349	Samsung_S10	AP_1			Authentication, SN=2466, FN=0, Flag
62	45.391504	AP_1	Samsung_S10			Authentication, SN=3444, FN=0, Flag
63	45.392572	Samsung_S10	AP_1	9c3cd31dffc71e2a0fcfcd7b98a94839		Association Request, SN=2467, FN=0,
64	45.393148	AP_1	Samsung_S10			Association Response, SN=3445, FN=0
65	45.398606	AP_1	Samsung_S10		9c3cd31dffc71e2a0fcfcd7b98a94839	Key (Message 1 of 4)
66	45.434556	Samsung_S10	AP_1	9c3cd31dffc71e2a0fcfcd7b98a94839		Key (Message 2 of 4)
67	45.438193	AP_1	Samsung_S10			Key (Message 3 of 4)
68	45.449602	Samsung_S10	AP_1			Key (Message 4 of 4)
69	55.663804	Samsung_S10	AP_2			Authentication, SN=2506, FN=0, Flag
70	55.664380	AP_2	Samsung_S10			Authentication, SN=2038, FN=0, Flag
71	55.665382	Samsung_S10	AP_2	1914c2313f0bbda1913cd56f7b152b73		Reassociation Request, SN=2507, FN=
72	55.666572	AP_2	Samsung_S10			Reassociation Response, SN=2039, FN
73	55.675614	AP_2	Samsung_S10			Request, Identity
99	56.403471	AP_2	Samsung_S10			Success
100	56.405284	AP_2	Samsung_S10		555cf381c5e14ab54bb6f7b0e5a0acfd	Key (Message 1 of 4)
101	56.418278	Samsung_S10	AP_2	1914c2313f0bbda1913cd56f7b152b73		Key (Message 2 of 4)
102	56.423474	AP_2	Samsung_S10			Key (Message 3 of 4)
103	56.435439	Samsung_S10	AP_2			Key (Message 4 of 4)

Picture 4. Packet capture for step 4 to 5.

Explanation

- Back at step 4.
    - The client has been disassociated and its NIC has been turned off. But since it associates to the same AP as it left, AP1, and both stations (client and AP1) has the cached PMKID it goes directly to the 4-way handshake
- Step 5
    - This is the odd situation. Normally we would have expected the client send the PMKID from the previous association to this AP2, step 2. But it sends an unknown PMKID never used before. AP2 don't know this PMKID and initiates a full 802.1X/EAP authentication.
      To shorten the picture the EAP-process are hidden (frame 76-99).
    - This happens also if the client disassociates at AP1 and associate later to AP2 and the clients NIC has been turned off.

Conclusion

During testing on a WPA2-Enterprise (802.1X/EAP) WLAN with default setup on Mist APs, roaming not enabled, the Samsung S10 has another association behaviour than all the other tested clients.
Most clients do full 802.1X/EAP-authentication during association and roaming (reassociation) between APs, while the Samsung S10 very often skips the 802.1X/EAP-authentication and goes directly to the 4-way handshake.

The re-association behaviour is what we are knows as PMK Caching.
The association behaviour is also PMK Caching and can be used in the lifetime of the PMKSA (PMK Security Association).

So it seems there are very few clients on the marked that supports PMK Caching, fewer than many expects.

The odd behaviour in step 5 seems to be a small bug in the Samsung operating system.

What is the benefit of PMK Caching during association and re-association

The main benefit of PMK Caching is to avoid the 801.1X/EAP process where the client talks to a Radius server.
During my testing this takes 800ms.
Avoiding this, especially in crowed areas, would improve performance of WiFi networks
FILS Authentication is a method to improve this behaviour, but it is very seldom implemented