Wi-Fi Tracking and MAC Address Randomization

AMJAD ABU-MAHFOUZ,

RISHAB DHAMIJA AND

WAHAB SAFDAR

INTRODUCTION

- - all 802.11 devices come with interface
- -interface comes with MAC address
- -MAC address is 48-bit and UNIQUE!!

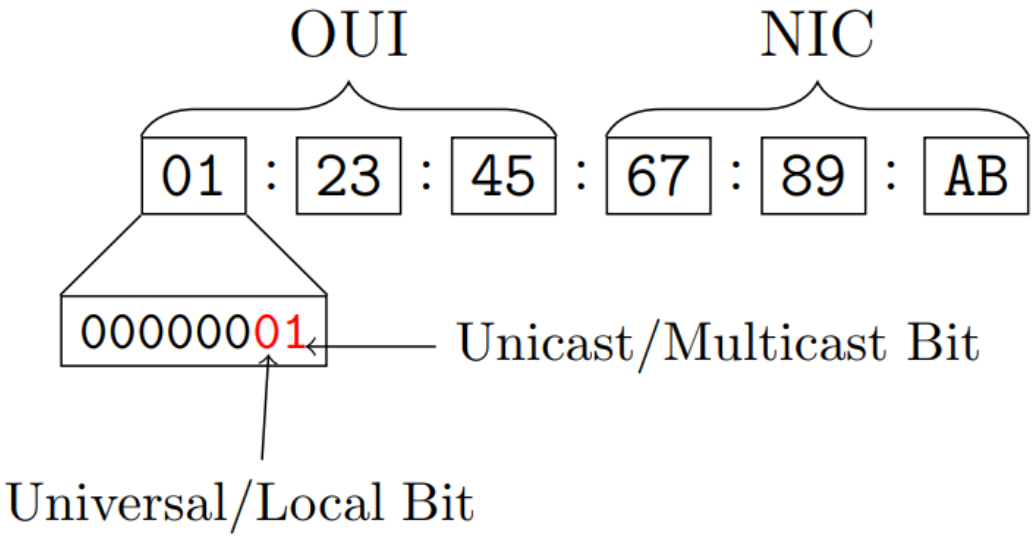
- - IEEE : Institute of Electrical and Electronics Engineers
- - IEEE 802 is a set of LAN protocols
- - IEEE 802 tells what MAC & layer2 protocols to make a WLAN

INTRODUCTION- CONTINUED

- -IEEE sells MAC address blocks
- -MAC addrress blocks purchased act much like subnet addresses
- - 3 bytes (manufacturer ID **like subnet network addr.) + 3 bytes (**host ID)
- -*** IEEE wont sell unless agreed to make MACs unique

MAC ADDRESS

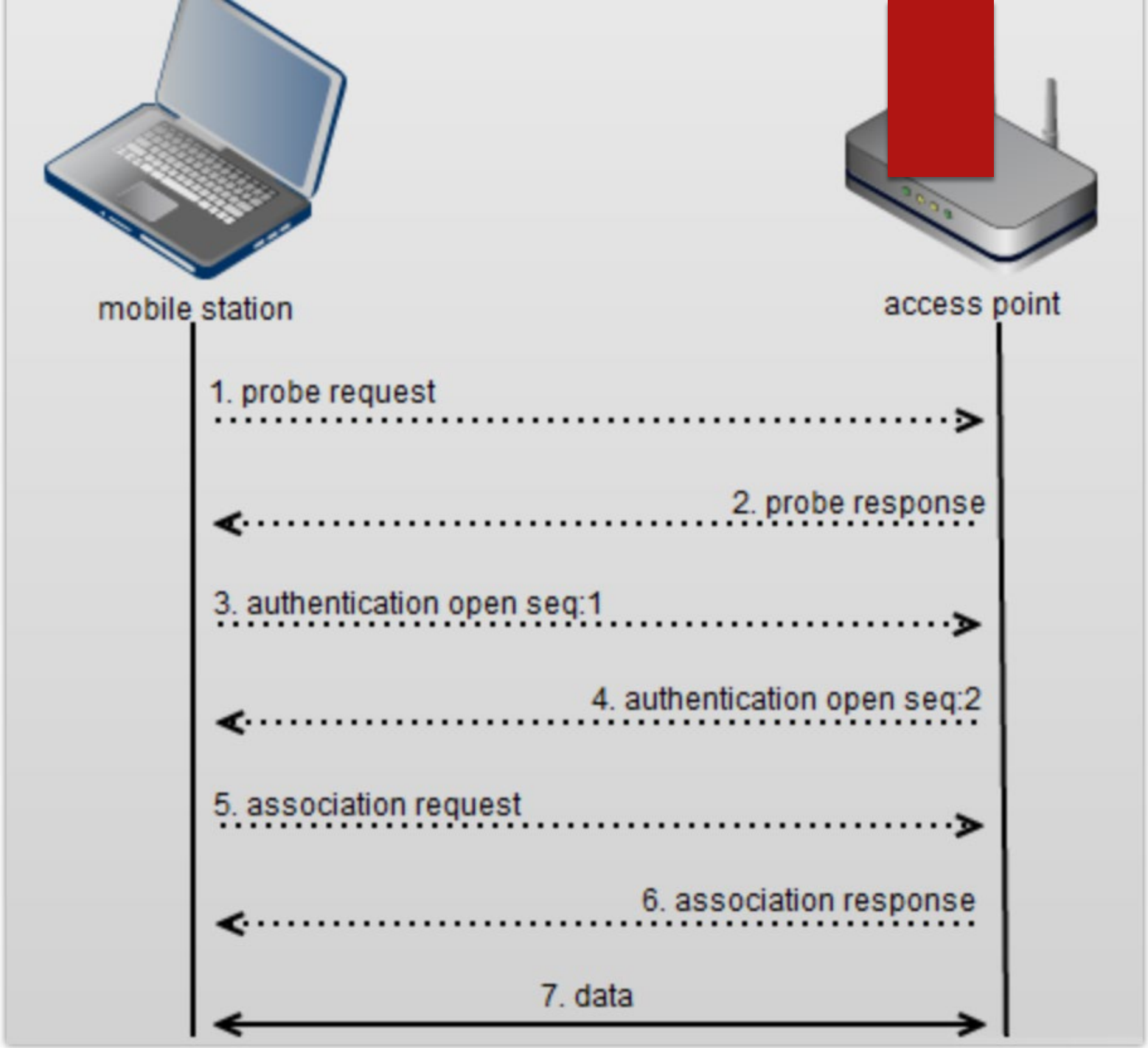
- - the important bit is the universal/local bit (found in OUI or "manufacturer ID" part of MAC)
- when set to ....
- -locally assigned MAC : MAC is treated as non persistent (reusable) & non unique
- -Globally assigned MAC : MAC is unique (normal version)
- -***IMPORTANT*** : locally assigned MAC can be used for MAC randomization



RISKS/THREATS

- - locally assigned MAC : hard to track you
- - Globally assigned MAC : you can easily be ID'd/tracked
- - ***IMPORTANT***: when a device needs to connect to an access point (Gateway) ...
- -it broadcasts "probe frames" to search for surrounding access points
- -probe frames contain globally assigned MAC as source MAC : unsecure!!
- - Android & Apple iOS offer security by letting you broadcast probe frames using locally assigned MAC

Probe/Request Authentication Association cycle



MAC Randomization Schemes

- **Linux** supports MAC address randomization, and lets the driver or firmware generate per-burst random MAC addresses.
- . As a result, most Linux devices change their MAC address at most every few bursts. The default duration for a random MAC address is 60 seconds.
- MAC address randomization is supported by iOS since version 8. Randomization is limited to probing and only happens under specific conditions: the device has to be unassociated and in sleep mode.
- Windows 10 changes the MAC address when the device connects or disconnects from a network, and when it is restarted. For such implementations, tracking is trivial since the device identifier does not change during a tracking session.
- **with current implementation of randomization, the same MAC address is used over at least one burst and can cover multiple consecutive bursts.**

Vulnerability 1: Abusing the timing of 802.11 probe frames

- the wireless driver of Wi-Fi enabled devices can be fingerprinted using the inter-arrival time of the probe requests
- This is used to group together frames coming from the same device although they use distinct MAC addresses.
- These signatures are consistent over time and can be used as a pseudo-identifier to track devices.

Vulnerability 1 : countermeasure

- First, changing the MAC address more often, e.g. every burst or every frame, has the potential to reduce the trackability of devices, as the amount of information for fingerprinting will be limited to a few frames.
- Since this attack relies on temporal pattern, a simple countermeasure would be to break those patterns by introducing some random delays between probes and between bursts.

Vulnerability 2 : UUID–E Reversal

- **UUID-E is derived from a device's global MAC address,** and by using pre-computed hash tables an attacker can simply lookup the UUID-E from the table and retrieve the global MAC address
- THIS UUID-E DOES NOT CHANGE DESPITE RANDOMIZATION IN PLACE

Vulnerability 2 : Analysis

- Basic requirement for this attack to work : WPS attributes need to be present within the request (especially the UUID-E) attribute. (Motorola devices , especially Nexus series fall into this category)
- NOTE: THE TIME THIS ATTACK CAME INTO EXISTENCE , RANDOMIZATION WAS NOT IN USE.
- Most of the devices using Google CID do not transmit WPS attributes (Android) .
- Therefore, no information about manufacturer and model number available for those devices.
- Moreover, without the WPS attributes, UUID-E Reversal can't be applied to retrieve Global MAC address.
- CHEERS FOR ANDROID USERS !!! AT LEAST IN THIS REALM FOR SURE…..

SKIPPING iOS … NOT FAIR AT ALL!!

- Upon the release of iOS 8.0, Apple introduced MAC address randomization.
- Initial thought for iOS → Apple would use a OUI or CID like other manufacturers and simply randomize the least significant 24 bits of the MAC address.
- ACTUALLY, MAC addresses randomly generated by iOS devices do not share any common prefix .
- IN FACT, they appear to be completely random, including the 24 OUI bits, except for the local bit which is always set to 1 and the multicast bit which is set to 0.
- Is Apple freely making use of address space that other companies have paid for ??

Vulnerability 3 : Global Probe Requests

- Flaw across Android devices → , the inexplicable transmission of the global MAC address in tandem with the use of randomized MAC addresses .
- Between probe requests, **the sequence numbers increase predictably** so an entire series of random addresses can be linked with a global address by just following the chain of sequence numbers.
- ANDROID USERS BEWARE !!!

Vulnerability 4 : Karma Attack

- The current versions of iOS and Android randomization policies have eliminated the vast majority of cases where a directed probe is used.
- Directed Probe → probe request containing a specified SSID that the device wishes to establish a connection (a previously known or configured SSID
- Broadcast Probe → a probe request that solicits a response from all APs in range.
- LET'S TAKE ADVANTAGE OF THIS … KARMA ATTACK.

Vulnerability 4 : Continued

- Karma-based attacks work by simulating an access point that a device prefers to connect to.
- Follow-on-consequences → man-in-the-middle attack.
- Since, we are only interested in retrieving the Global MAC address, we require only a single Authentication frame transmitted from the target device. (how about using top SSIDs for the AP ??)
- Active baits for this attack ?? Saved networks from previous ad-hoc connections in the network's list of the target machine force it to send directed probes with specific SSID in it.

Vulnerability 5 : Control Frame Attack

- NOT EXACTLY A VULNERABILITY, BUT A CONSEQUENCE.
- Lets target something OS vendors have no control over → 802.11 chipsets.
- Request-to-send (RTS) / Clear-to-send(CTS) are available in the 802.11 specification as part of CSMA/CA.
- A node wishes to transmit data → send a RTS alert → "all other nodes to not transmit to avoid collision". → Destination node respond with CTS → transmission node receives permission to solely communicate with it.
- However, CTS response provides source MAC address despite randomization. Therefore, once the global MAC is known, send RTS to the client machine and you are good to go! OS vendors can't do anything here.

Benefits Of Mac Randomization

- MAC address randomization follows the idea to use disposable interface identifiers in order to improve users' privacy. In practice, this implies that probe requests no longer use the real MAC address of the device.
- For example, a new MAC address can be used for each scan iteration, where one scan iteration consists of sending probe requests on all usable channels.
- By doing so WIFI tracking can be stopped in the event someone is probing WIFI and the MAC address can lead to tracking. But with MAC randomization new MAC address will be associated.

Real Life Example

- Four scholars from the US Naval Academy say they've managed to track 100% of all test smartphones, despite the devices using randomized MAC addresses.
- The technique worked across all tested manufacturers, and the researchers say this was possible because of a previously unknown flaw in the way wireless chipsets handle low-level control frames.
- By sending RTS frames to IEEE 802.11 client devices, to extract a CTS response message derives the true global MAC address of a device.

Real Life Example

- By sending a RTS frame to the global MAC address of a device performing randomization the target device responded with a CTS frame.
- CTS frame, having no source MAC address, confirmed as a response to the attack based on the fact that it was sent to the original, crafted source MAC address, once the global MAC address is known, that device can be easily tracked just as if randomization were never enabled.

Questions

1) Based on the iOS scheme of assigning random MAC addresses, what would be the total bit space they actually have control over ??

Answer : removing the 7th local bit and 8th multicast bit, they are free with 46 bits!!.

2) Can you describe a way Karma attack can be used as man-in-the-middle attack ?? Just give some ideas.

Answer : you simulate an AP and sit in between the sender and the actual AP. Send the original machine probe responses with the actual SSID the sender wished to connect to and he will think you are actual AP.

3) What makes control frame attack different from rest of the attacks we have covered above ??

Answer : unlike other attacks, it went to the lower level of 802.11 chipsets which is outside the reach of OS vendors.