A Study of MAC Address Randomization in Mobile Devices and When it Fails

PETS 2017 - to appear

Jeremy Martin, Travis Mayberry, Collin Donahue, Lucas Foppe, Lamont Brown, Chadwick Riggins, Erik C. Rye, and Dane Brown

June 27, 2017

Outline

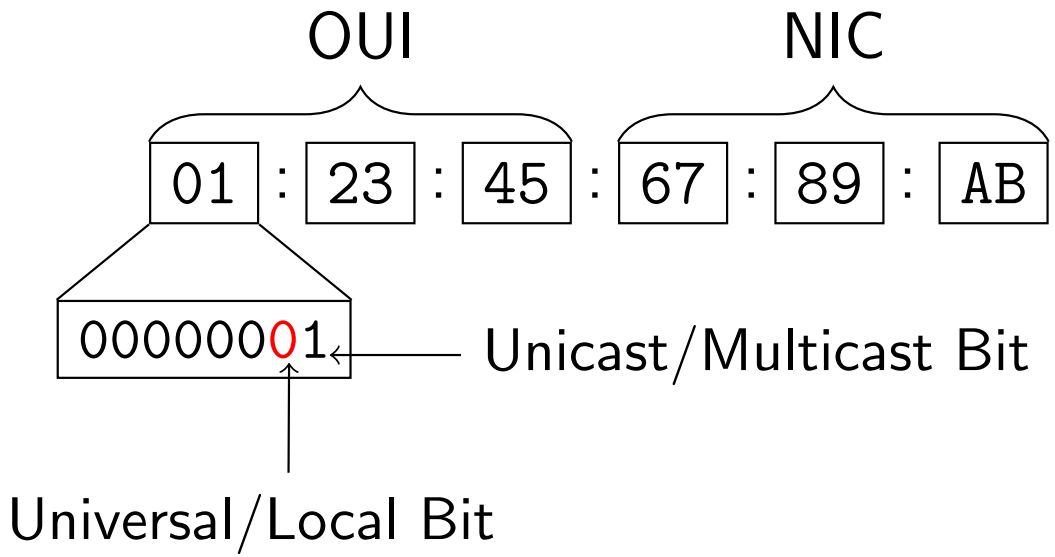
- Every 802.11 radio on a mobile device possesses a MAC address that is globally unique
- The MAC address is a crucial part of WiFi communication, being included in every link-layer frame
- **This unfortunately poses a glaring privacy problem!**
- To address this problem, some modern mobile devices make use of temporary, randomized MAC addresses global address

In this work we evaluate the effectiveness of various deployed MAC address randomization schemes

Abstract

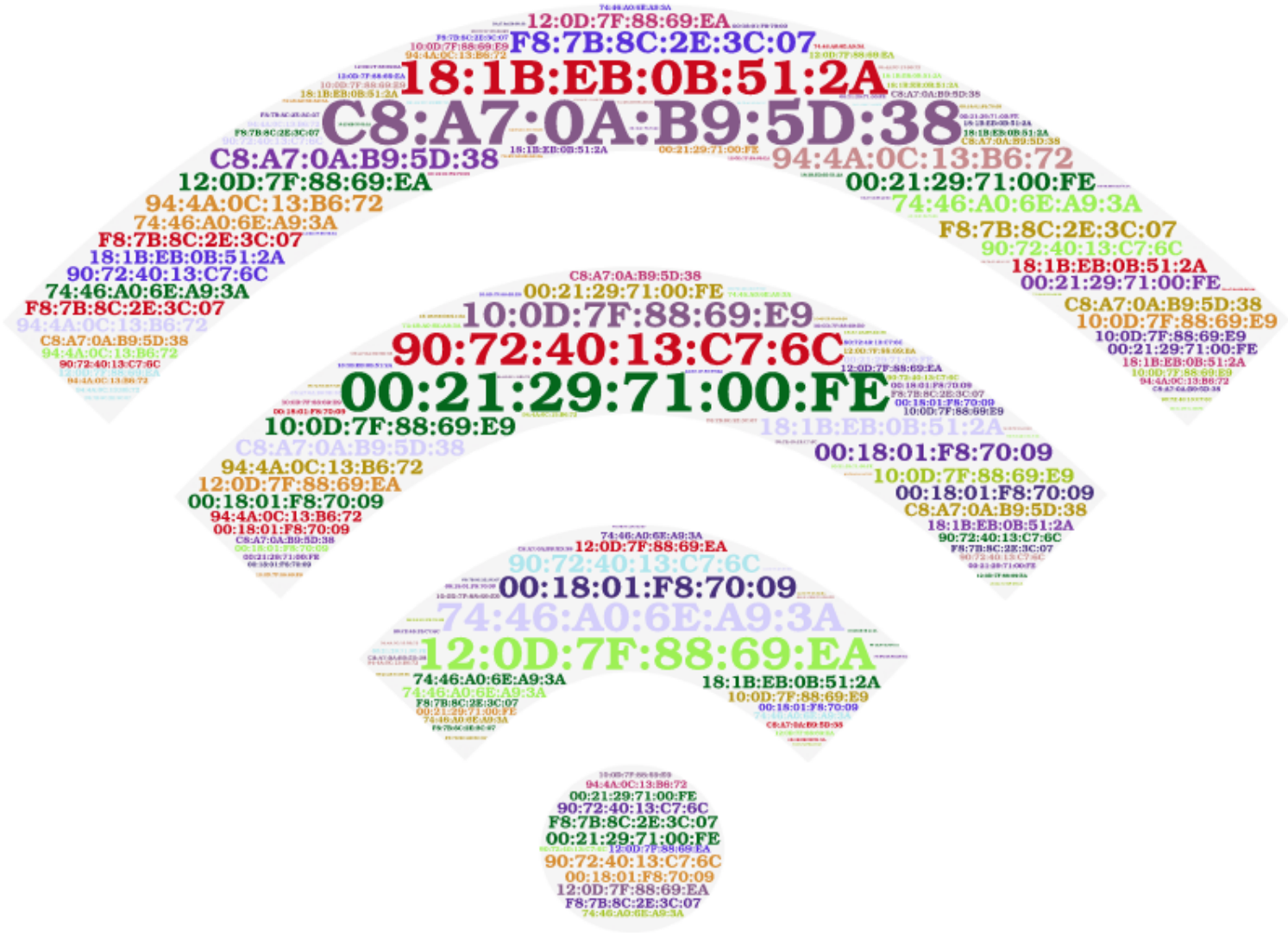
- The first wide-scale study of MAC address randomization in the wild
- Detailed breakdown of randomization techniques by operating system, manufacturer, and model of device
- Identify multiple flaws in these implementations which can be exploited to defeat randomization

- Traditionally, devices perform active scanning while in an unassociated state
- If a device uses its unique MAC address then it is effectively broadcasting its identity at all times
- To combat this privacy concern, both Android and Apple iOS allow for devices in a disassociated state to use random, locally assigned MAC addresses when performing active scans
- Since the MAC address is now random, users gain a measure of anonymity up until they associate with an AP

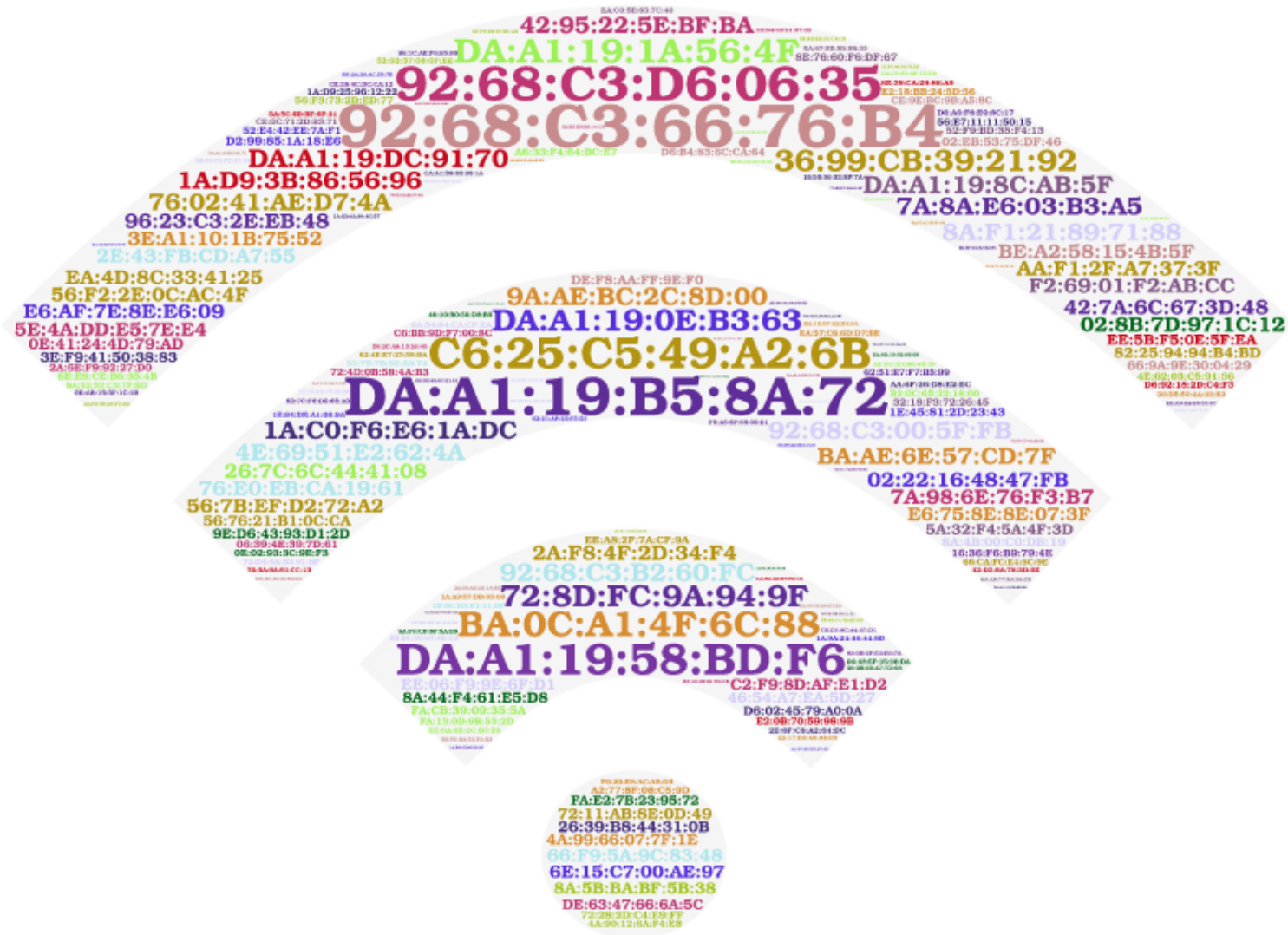


While the IEEE does not specifically provide guidance for the structure of Randomized MAC addresses, the *local* bit provides a logical implementation solution commonly employed by mobile OSes.

- 802.11 corpus
  - Jan 2015 - Dec 2016
  - ∼600 GBs / 9,000 packet captures
  - As per IRB, we analyze only management frames and unencrypted mDNS packets
- Identify randomization
- Identify and evaluate implementation flaws



We start with all unique source addresses, there sure is a lot!



By using the *local* bit we reduce our dataset in half

Next we remove P2P and other *service*-based addresses

Using a variety of techniques we categorize the remaining data

Category	# MACs
Corpus	2,604,901
Globally Unique	1,204,148
Locally Assigned	**1,400,753**

Category	# MACs
Locally Assigned	**1,400,753**
Randomized	**1,388,656**
Service	4,371
Malformed	6,895
Unknown	831

Category	# MACs
Randomized	**1,388,656**
Android: `DA:A1:19` (WPS)	8,761
Android: `DA:A1:19`	43,924
Android: `92:68:C3` (WPS)	8,961
iOS	1,326,951
Windows 10 / Linux	59

Android: DA:A1:19

- Google owned CID, hard-coded into *config.xml* file
- WPS-derived attributes:
  - Provide granular manufacturer and model details
  - Provide data allowing for derandomization attacks
- Subset of devices do not transmit useful WPS data
  - Required additional level of analysis

Android: DA:A1:19 (WPS) – Device Breakdown

Manufacturer	Total Devices	Model Diversity
Huawei	1,708	11
Sony	277	†23
BlackBerry	234	‡4
HTC	108	2
Google	13	2
LG	1	1

† All Xperia variants
‡ All Priv variants

- Alarmingly low manufacturer and model diversity!
  - Samsung, Motorola, and LG poorly represented

Android: DA:A1:19 (no WPS) – Device Breakdown

Device	% of no WPS
LG Nexus 5X Google Pixel	57.7%
LG G5 LG G4	18.5%
OnePlus 3 Xiaomi Mi Note Pro	2.0%
Huawei Sony	.2%
Cat S60	2.6%
Composite	12.2%
Unknown	6.8%

- Samsung and Motorola still conspicuously absent...

Android: 92:68:C3

- Hard-coded into *config.xml* file, replaces Google CID
- Derived from WPS attributes:
  - The prefix is used singularly by the Motorola Nexus 6
  - 849 distinct devices observed

Android: Where are Motorola and Samsung??

- Non-standard randomization?
- Identify OUIs with unusually high occurrences within individual packet captures
  - Various Motorola OUIs observed with an abnormally high percentage of the unique addresses in a packet capture
  - Various Moto G, E, and Z series confirmed using non-locally assigned random MAC addresses

Android: Where is Motorola and Samsung??

- We never observed Samsung devices performing MAC address randomization
- Samsung uses their own 802.11 chipsets, so it is possible that chipset compatibility issues prevent implementing randomized MACs addresses
- In our lab setting we confirm Samsungs lack of randomization when tested against a wide range of Samsung models and OS versions

iOS

- At first iOS randomization was difficult to positively identify
- With the release of iOS 10 a unique field was added that allows for trivial identification
- Universally adopted across iOS devices running iOS 8.0^{+}
- Randomizes across the entire 2^{46} bit space

Adoption Rate

- The overwhelming majority of Android devices are not implementing the available randomization capabilities built into the Android OS
- No effort by an attacker is required to target these devices
- Furthermore, the lack of adoption allows for simpler identification, effectively simplifying the problem set

Global Probe Request

- Global MAC address used in tandem with randomized MAC addresses!! $\rightarrow$ Effectively circumventing randomization
- Observed across the gamut of Android devices †
- Additional global probe requests sent when:
  - User activates the screen for any function
  - Incoming phone call, allowing for side channel stimulation

In effect, this renders randomization moot, eliminating the privacy countermeasure all together.

†With the exception of the Cat S60 smartphone

UUID-E Reversal

- Flaw caused UUID-E construction, where the MAC address is used as an input variable with a non-random seed value
- Using pre-computed hash tables, retrieving the global MAC address requires only a simple lookup query
- Our knowledge of randomization adoption rates allowed for improved efficiency of hash tables computation
- $\sim$29% of Android dataset contained WPS attributes
- iOS invulnerable to this attack
- Achieved a $\sim$99.96% global MAC address retrieval rate

Device Signature

$Sig_G =$ `0,1,50,3,45,221(0x50f2,8),htcap:012c,htagg:03,htmcs:000000ff`

- Borrowed from related work we create quasi unique device signatures mapped from probe request frame elements
- *Bins* device types, clearing out the *noise*
- Analysis implemented in parallel with sequence number inspection
- When used with other flaws (global probe request) allows for both tracking and derandomization

Association/Authentication Frames

- Association and authentication frames from iOS and Android devices use the global MAC address
- Using device signatures and sequence numbers we correlate the randomized addresses with the device's global address
- Requires that a device attempts to associate to a network during collection

Karma Attack

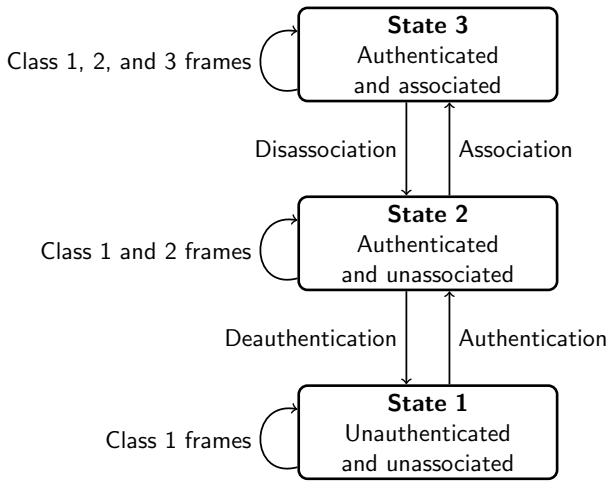
- Effectively an active attack extension of the passive Association/Authentication flaw
- Build on previous work by specifically evaluating randomizing devices
  - Increased attack vector caused by WiFi data-offloading settings settings
  - $\sim$17% of directed probes caused by preconfigured mobile provider settings
  - Expect this number to grow with the growing adoption of Hotspot 2.0, EAP-SIM, and EAP-AKA

Control Frame Attack

Our premise: can we force a device performing MAC address randomization to respond to frames targeting the global MAC address?

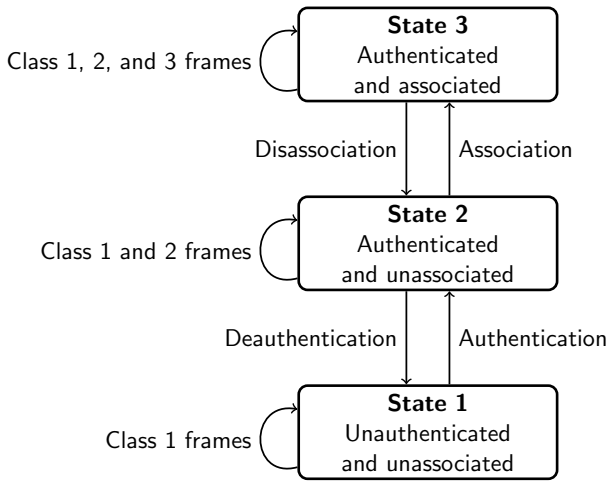
Control Frame Attack

Our premise: can we force a device performing MAC address randomization to respond to frames targeting the global MAC address?



Control Frame Attack

Our premise: can we force a device performing MAC address randomization to respond to frames targeting the global MAC address?



Control	Management	Data
RTS	Probe Request	Frame w/DS bits false
CTS	Probe Response	
Ack	Beacon	
CF-End	Authentication	
CF-End+CF-Ack	Deauthentication	
	ATIM	

RTS/CTS

- Sending a RTS frame to the global MAC address of a device performing randomization elicits a CTS frame response
- If the global MAC address is known, that device can be easily tracked just as if randomization were never enabled
- Android and iOS are both susceptible
- This leads us to believe that RTS/CTS responses are not a function of the OS, but of the underlying 802.11 chipset

Derandomization Technique Results

Randomization Bin	Global MAC Address Probe Request	UUID-E Reversal	Auth/Assoc Frames	Hotspot 2.0 - Karma Attack (*Active*)	RTS Attack (*Active*)
DA:A1:19 with WPS	√	√	√	√	√
DA:A1:19 w/o WPS	√	×	√	√	√
92:68:C3 with WPS	√	√	√	√	√
Motorola (No local bit)	√	×	√	√	√
Apple iOS	×	×	√	√	√

- MAC address randomization policies are neither universally implemented nor effective at eliminating privacy concerns
- The table depicts the diversity of presented attacks, across the spectra of randomization schemes and OSs

- Android devices are susceptible to the spectrum of passive and active derandomization techniques
- Samsung devices do not conduct randomization at all, failing to provide a modicum of identifier obfuscation
- Conversely, iOS devices, while broken for some edge cases, require specific network interaction and/or active attacks for defeating randomization implementations

- To be truly effective, randomization should be universally adopted
- A universal policy must include at minimum, rules for randomized MAC address byte structure, 802.11 IE usage, and sequence number behavior
  - A detailed granular list of example policy rules are provided in the full paper