Poster: Defending Wi-Fi Network Discovery from Time Correlation Tracking

Federico Cifuentes-Urtubey, Robin Kravets, Deepak Vasisht
University of Illinois at Urbana-Champaign
{fc8, rhk, deepakv}@illinois.edu

ABSTRACT

To prevent tracking a Wi-Fi device based on its MAC address, several operating systems have adopted MAC address randomization to conceal its factory-assigned address. This feature benefits users when their devices scan for networks, but a flaw arises when timing between transmissions stays consistent despite MAC address randomization in use. We present a defense mechanism, implemented with the Netlink library, against a time correlation attack for probe request packets on Wi-Fi devices. We show how adding random jitter to probe request transmissions renders a timing correlation attack infeasible to track devices during network discovery.

CCS CONCEPTS

• **Networks** → **Link-layer protocols**; **Mobile and wireless security**.

KEYWORDS

MAC address randomization, Privacy, Probe requests, Wi-Fi

1 INTRODUCTION

Public Wi-Fi is available almost everywhere, spanning coffee shops to hotels and airports visited while traveling. Having this accessibility has made Internet connectivity more convenient, but it also threatens users being tracked if attackers can identify their mobile devices. This vulnerability comes from the fact that Wi-Fi devices constantly scan their environment to find access points (APs) to connect to while broadcasting their unique identifier, a MAC address, unencrypted. The resulting broadcast enables an adversary to sniff Wi-Fi packets to track the device transmitting them. Simple encryption of the probe packets is not an option since probes must be transmitted unencrypted so nearby APs can reply. As a countermeasure, several operating systems have adopted MAC address randomization to disconnect one network discovery event from the next. Unfortunately, MAC randomization does not consider full use of the MAC address in network discovery and so fails to protect a device from being tracked due to temporal patterns in packets within a network discovery event [2].

To prevent timing-based tracking attacks on Wi-Fi discovery, we present a defense that breaks transmission patterns in and across discovery events. Within a discovery event, multiple probe requests are sent on the same channel with predefined timing between each probe [2]. By measuring the time deltas between probes, a temporal fingerprint can be created to isolate and track the device [3], nullifying the benefits from MAC randomization. To recover these benefits, our approach introduces a randomized probe delay on each frame in a discovery event. By breaking deterministic timing between probes, attackers can no longer fingerprint a device's discovery event. Based on our experiments, randomizing the probe delays within a discovery event prevents the timing-based tracking algorithms from finding matching probe behaviors. Additionally, our approach randomizes the time between discovery events, however, we focus on the timing within discovery events.

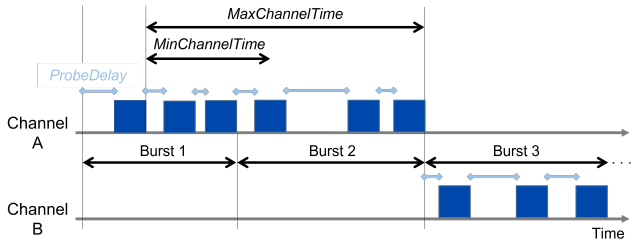

Figure 1: Randomizing *ProbeDelay* on each frame breaks the timing pattern between probes. After *MaxChannelTime*, a device will scan on the next channel.

2 THREAT MODEL

In a public Wi-Fi network, users can connect to non-malicious APs configured with open authentication. To support MAC randomization, the user's device has the permissions to change its own MAC address. The adversary's main goal is to attribute probe requests to a specific device in the environment even when devices use MAC randomization. A passive adversary is able to scan the environment for Wi-Fi traffic one channel at a time and utilize any data in the clear when a device transmits Wi-Fi frames. However, the adversary does not have access to user devices or the AP. These scans enable the adversary to read link layer packet headers and identify fields that could allow a device to be traceable, such as the MAC address. Additionally, the adversary can calculate time deltas between probe requests transmitted by devices to look for identifying patterns.

3 PATTERNS IN WI-FI DISCOVERY EVENTS

Probe requests in network discovery have distinguishable patterns despite using MAC randomization because of inconsistencies between implementations of the feature [1]. For example, a device may change only the last three bytes of its MAC address while another device only changes the locally administered bit in the MAC address. Differences between MAC randomization behaviors such as these allow an adversary to identify devices when probe requests are transmitted during a discovery event.

Discovery events occur when a device actively scans its environment for APs. The device configures each discovery event to set

MobiSys '22, June 25-July 1, 2022, Portland, OR, USA

ACM ISBN 978-1-4503-9185-6/22/06.
https://doi.org/10.1145/3498361.3538799

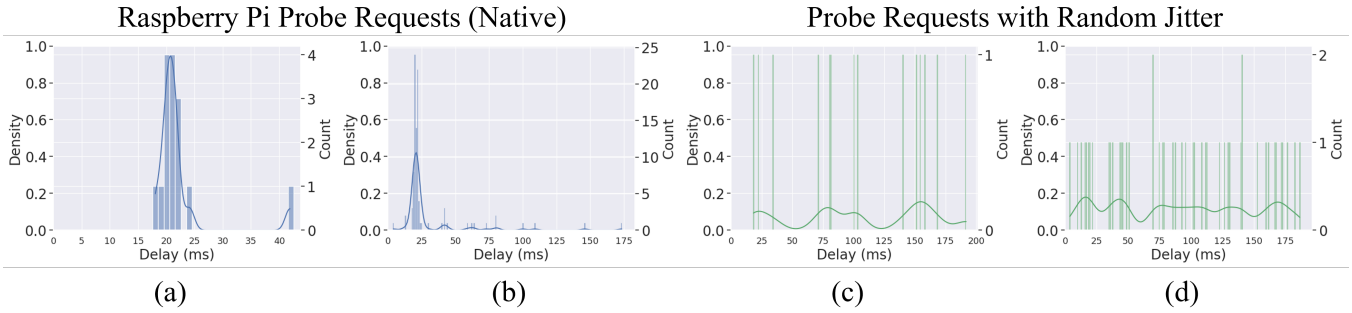

Figure 2: IFAT distributions from (a) Raspberry Pi, 1 minute capture (b) Raspberry Pi, 20 minute capture (c) Netlink with random jitter, 1 minute capture (d) Netlink with random jitter, 20 minute capture

the number of probes transmitted, time to wait before sending a probe request (*ProbeDelay*), channels to scan, and timers for how long it stays on a channel (*Min/MaxChannelTime*). By default, APs broadcast beacons every 100ms, so a device may send multiple bursts within a *MaxChannelTime* of 100ms to proactively search for them. If the device does not receive a probe response within *MinChannelTime*, it will start scanning the next channel. As probes are transmitted, they use the same *ProbeDelay*, which causes the same number of bursts to appear on a given channel. These configuration parameters create patterns because they determine the delay for each probe, how many bursts of probes are transmitted on a channel, and how many probes are sent in each burst. MAC randomization implementations create additional patterns because of variance in how often they change MAC addresses and when transmitting probe requests containing changed addresses [1]. To develop a time-based defense while MAC randomization is in use, adding jitter to *ProbeDelay* on each transmission becomes necessary to hide the pattern embedded in the IFATs.

4 ANONYMIZING WI-FI DISCOVERY EVENTS

In a discovery event, a device transmits several probe requests on one channel in a burst with the same probe delay applied to each frame. With a passive adversary observing the probes, they can measure the delay between probes as the Inter-frame Arrival Time (IFAT). Then, the adversary uses IFATs to correlate probes with different MAC addresses to the device broadcasting those probes. The same probe delay results in consistent IFAT measurements. Eliminating the resulting transmission pattern requires MAC randomization with a randomized delay (i.e., adding jitter) on a per packet basis. Figure 1 illustrates this discovery modification to affect the *ProbeDelay* parameter, where each probe is assigned a different *ProbeDelay* value within each burst. The device will remain on the channel for at least *MinChannelTime*, and if it does not receive a probe response within that time, then scanning advances on next channel. After spending *MaxChannelTime* on the current channel, the device starts to probe on a different channel with randomized *ProbeDelay* on each frame. To measure similarity between modified discovery events, a capture of probe requests is fed into a timing attack algorithm [2] to create sets of MAC address aliases. The algorithm measures IFATs for each MAC address separated by a 200ms discovery event window. Then, it finds the mean and median of the probes' IFATs for each MAC address as a signature. If two MAC addresses have similar metrics, it indicates a traceable transmission behavior.

Results: We implement our design with the Netlink library for the Linux kernel and evaluate the design with a time correlation algorithm [2]. The Netlink program transmits probe request frames with random jitter on each frame while a monitor records the arrival times and calculates IFATs. Figure 2 visualizes the IFAT measurement counts and distributions for probing behaviors of a Raspberry Pi and modified probe request transmissions for packet captures spanning 1 minute and 20 minutes. Distributions for both time windows show that a Raspberry Pi transmits probe requests with an IFAT median of 21ms while the modified probe requests for both time windows show a more uniform distribution with medians of 101ms (1min) and 92ms (20min). Since modified probe requests transmit with random delays, the amount of probes per burst changes to make IFAT measurements more variable. Analyzing IFATs from the Raspberry Pi's discovery events allowed the time correlation attack to successfully identify it 88% of the time in the 1min capture and 75% of the time in the 20min capture. The decrease is due to channel contention from other devices, further delaying the probe transmissions. With the modification, the time correlation attack was unable to match jittered frames coming from the same device for both time windows. Based on experimental results, adding randomized jitter to probe request transmissions during discovery events is a feasible defense against time correlation attacks on Wi-Fi devices.

REFERENCES

[1] Ellis Fenske, Dane Brown, Jeremy Martin, Travis Mayberry, Peter Ryan, and Erik Rye. 2021. Three Years Later: A Study of MAC Address Randomization In Mobile Devices And When It Succeeds. *Proceedings on Privacy Enhancing Technologies* 2021, 3 (July 2021), 164–181. https://doi.org/10.2478/popets-2021-0042

[2] Célestin Matte, Mathieu Cunche, Franck Rousseau, and Mathy Vanhoef. 2016. Defeating MAC Address Randomization Through Timing Attacks. In *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec '16)*. 15–20. https://doi.org/10.1145/2939918.2939930

[3] Mathy Vanhoef, Célestin Matte, Mathieu Cunche, Leonardo S. Cardoso, and Frank Piessens. 2016. Why MAC Address Randomization is Not Enough: An Analysis of Wi-Fi Network Discovery Mechanisms. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security (ASIA CCS '16)*. 413–424. https://doi.org/10.1145/2897845.2897883