Truncate after Preamble: PHY-based Starvation Attacks on IoT Networks

Stefan Gvozdenovic
tesla@bu.edu
Boston University
Boston, MA

Johannes K Becker
jkbecker@bu.edu
Boston University
Boston, MA

John Mikulskis
jkulskis@bu.edu
Boston University
Boston, MA

David Starobinski
staro@bu.edu
Boston University
Boston, MA

ABSTRACT

We present and evaluate Truncate-after-Preamble (TaP) attacks, whereby a receiver cannot decode an incoming signal despite good channel conditions. In a TaP attack, the attacker announces a large payload length using a standard preamble and packet length field, but omits to transmit the payload. We implement the TaP attack on a SDR platform, and evaluate the effectiveness of the attack on five Zigbee and seven Wi-Fi devices sold by different manufacturers. We show that all of the Zigbee devices are vulnerable to the attack, while the Wi-Fi devices are vulnerable to the attack to varying degrees. Chiefly, we show that an attacker can cause over 90 % packet loss on a Zigbee or Wi-Fi channel, using respectively six or five orders of magnitude less energy than a constant jammer would. Finally, we present several methods, with different degrees of sophistication, for detecting the attacks.

CCS CONCEPTS

• **Security and privacy → Denial-of-service attacks**; **Mobile and wireless security**.

KEYWORDS

Internet of Things, Denial of Service, Physical Layer, Software-defined Radio,

ACM Reference Format:
Stefan Gvozdenovic, Johannes K Becker, John Mikulskis, and David Starobinski. 2020. Truncate after Preamble: PHY-based Starvation Attacks on IoT Networks. In *13th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '20), July 8–10, 2020, Linz (Virtual Event), Austria.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3395351.3399356

1 INTRODUCTION

Numerous IoT applications are emerging thanks to the low cost of IoT devices. For instance, the multi-protocol chip EFR32MG1 [16] costs a few dollars in high volume. The IEEE 802.15.4 and IEEE 802.11 protocol specifications play a prominent role in this regard. Indeed, many popular application-optimized IoT protocols are based on the IEEE 802.15.4 standard. This includes Zigbee, WirelessHART, 6LowPAN, Thread, and DotDot. Similarly, many IoT applications, such as WeMo, are based on IEEE 802.11 and its variants. It is expected that by 2024, the number of IEEE 802.15.4 and IEEE 802.11 annual device shipments will hit as much as 1 billion and 4 billion devices respectively [5, 31]. This connectivity will power, among others, building automation, medical applications, global supply chains, and factory floors.

Many IoT devices fulfil critical roles, ranging from smart building controls, such as boiler sensors, to medical devices, such as pacemakers and heart rate monitoring. Any attack on the availability of these devices may have dire consequences. Thus, ensuring the availability of mission-critical IoT devices is crucial for safety and security purposes.

Both IEEE 802.15.4 and IEEE 802.11 use a fixed preamble for all frame types. In these protocols, the preamble and starting frame delimiter are usually followed by an announced packet (frame) length. This packet length is a convenient way to tell the radio how many symbols it should expect to decode. The question arises whether any harm can be caused by sending false information about the packet length. In particular, are potential vulnerabilities dependent on the protocol specification, individual chipset vendor implementations, or a combination of both?

Under normal conditions, if a receiver relies on the packet length field, it should try to decode the announced packet until its end. Then, if the packet length information is inaccurate, the radio will report either a frame length error or a cyclic redundancy check (CRC) error. But what if an attacker announces a frame length and never follows up with an actual data payload?

In this paper, we introduce the *Truncate-after-Preamble* (TaP) attack, which leverages such "truncated packets" to trick receivers into listening to nonexistent transmissions at the expense of valid ones, causing starvation, i.e., making it difficult for devices to communicate in otherwise favorable channel conditions. The gravity of this attack lies not only in the starvation effect that it produces, but also in how it achieves this result. In contrast to a typical jamming attack, the TaP attack uses significantly less energy, which allows for low detection while still effectively denying service to any vulnerable devices within the attacker's communication range.

WiSec '20, July 8–10, 2020, Linz (Virtual Event), Austria

ACM ISBN 978-1-4503-8006-5/20/07…$15.00
https://doi.org/10.1145/3395351.3399356

Furthermore, unlike network allocation vector-based starvation attacks that are specific to Wi-Fi [2, 17], TaP attacks apply to a variety of wireless protocols and consume fewer bits and less energy (see Section 2).

Using a software-defined radio platform, we design and implement TaP attacks against five Zigbee and seven Wi-Fi devices. This setup allows one to create truncated packets, manipulate the packet length field in the preamble, and control the signal strengths. We introduce several metrics to quantify the potency of the attacks, both in terms of achieving effective denial of service and requiring low energy to mount. Specifically, we show that the attacker can cause over 90% packet loss on a Zigbee or Wi-Fi channel, using respectively six or five orders of magnitude less energy than a constant jammer would. We are unaware of any commercially available intrusion detection systems (IDS) on the market today that can identify and help counteract TaP attacks in a meaningful way. As a mitigation, we propose three methods to detect such attacks.

Our main contributions can thus be summarized as follows:

- We design and implement TaP attacks, using a USRP B200 SDR platform.
- We evaluate the behavior of five Zigbee and seven Wi-Fi devices, produced by different manufacturers.
- We show that all of the Zigbee devices are vulnerable to the attack.
- We show that Wi-Fi devices are susceptible to the attack to varying degrees.
- For all the Zigbee devices, we show that the attacks are effective even if the signal strength of the attacker is around 300 times lower than the target signal. In Wi-Fi, the signal's strength of the attacker can be around 30 times lower than the target's signal strength.
- We propose three complementary methods for detecting TaP attacks, based on regular consumer devices or more specialized tools.

The rest of this paper is organized as follows. In Section 2, we discuss related work. In Section 3, we introduce important concepts to this work, including the attack methodology, adversarial model and metrics. In Section 4 and 5, we present the experimental setup and results for IEEE 802.15.4 devices and IEEE 802.11 devices, respectively. In Section 6, we point out possible ways to detect the described attacks. We conclude the paper and discuss future work in Section 7.

2 RELATED WORK

In this section, we discuss works that share similar approaches to achieving denial of service (DoS) in wireless protocols through manipulation of low-level protocol features. However, none of them implements and evaluates an attack similar to the TaP attack.

2.1 IEEE 802.15.4/Zigbee and Z-Wave

O'Flynn [19] describes reactive 802.15.4 jamming, both through pulse jamming (24 µs duration) and interference messages (196 µs duration). Such reactive jamming requires adversarial power anywhere from -3 dB up to 6 dB compared to the target packet's power [19]. In contrast, TaP attacks do not directly jam the target's traffic, but rather trigger the target's reception and symbol decoding process, occupying their receive capability for up to the announced packet length. In contrast to reactive jamming attacks, a TaP attack can be effective even if the transmitted preamble has much lower power than the target packet, as long as the preamble is received before the actual target packet. In addition, the TaP attack is comparatively simpler to implement, as it neither requires strict timing as in [19, 30] nor power adaptation.

More recently, Ramsey *et al.* [25] demonstrate physical layer preamble manipulation as a means of fingerprinting and intrusion detection for Zigbee devices. This approach was subsequently further expanded to Z-Wave (ITU-T G.9959) devices by Hall *et al.* [8]. The TaP attack also makes use of physical layer packet manipulation, but in a different way and for a different purpose (i.e., starvation).

2.2 IEEE 802.11 (Wi-Fi)

There exists a rich literature on DoS and jamming attacks on Wi-Fi networks. Bellardo and Savage [2] implement a MAC layer virtual carrier-sense attack on commodity Wi-Fi IEEE 802.11a/b hardware to fake large network allocation vector (NAV) duration values in RTS/CTS/ACK frames.

Similarly, Negi and Rajeswaran [17] demonstrate RTS-based reservation attacks, which indiscriminately target all devices that can sense the malicious reservation packet. In case of such an RTS reservation attack, countermeasures exist in the form of a reservation revoke in the IEEE 802.11 DCF standard, which is implemented by sending CTS with a zero duration NAV. Finally, *Gu et al.* [7] suggest null data frames to form a number of different complex MAC layer attacks.

Unlike NAV-based attacks which target specific Wi-Fi frame types (e.g., RTS, CTS, and ACKs), TaP attack targets all frame types and different wireless protocols. Specifically, we demonstrate TaP attacks on both Zigbee and Wi-Fi devices.

All of the aforementioned works present attacks operating at the MAC layer, i.e., manipulating fields in the MAC header to reserve the channel for longer than a benign station typically would. In contrast, our approach targets the radio's receive state machine at the PHY layer. Specifically, we exploit the fact that, according to the IEEE 802.11 standard, the receiver state machine should wait until the end of the announced packet length in order to start receiving new packets again [12, pp. 2315-2318, 2415-2418]. Consequently, the TaP attack may vary in its effectiveness depending on the vendor and model of the target device's Network Interface Card (NIC), as it yields different results whether it strictly adheres to the standard or not. Given a vulnerable target device, the TaP attack is more power efficient than fake RTS/CTS packets since truncated packets are shorter in length.

Both Xu *et al.* [33] and Pelechrinis *et al.* [20] suggest that intelligent saboteurs can manipulate the back-off functionality (duration field in Wi-Fi MAC layer) to gain continuous access to the medium by choosing artificially small back-off values. This is related to TaP attacks except that we manipulate packet length fields (PHY layer) to primarily impact the PHY receive state machine. Additionally, Xu *et al.* [33] describe a deceptive jammer which transmits valid packets back-to-back, saturating the channel (as opposed to random noise or constant tone jammers). In contrast, our attack transmits only truncated packet without the data payload, significantly reducing the attacker's power, and making it a stealthier attack than back-to-back packet jamming approaches.

Rahbari *et al.* [23] suggest jamming a very short (<3 µs) part of the victim's preamble to induce errors in frequency offset (FO) estimation. This semi-targeted attack is efficient for starving a single device. However, it is far more complicated to implement than TaP since it requires microsecond-precision timing. Furthermore, it is semi-targeted since it has to guess the packet's identity by analyzing previous protocol semantics (RTS/CTS/ACKs). Once a time slot of a packet is guessed, its preamble is jammed before the actual packet's identity is determined with certainty (e.g. MAC address). While this can be considered a reactive semi-targeted jamming attack, the TaP attack denies the service indiscriminately to all vulnerable devices that receive the truncated packet. Hence, the power efficiency of our attack increases with the number of devices in range.

Ramsey *et al.* [24] apply a physical layer preamble manipulation scheme to fingerprint various Wi-Fi device types based on their response to non-standard preambles – a technique directly derived from their previous works on IEEE 802.15.4 [25]. Our attack differs in leveraging the effect of truncating a packet after announcing a valid packet length, but does not employ manipulated non-standard preambles. As in the work of Ramsey *et al.*, the effectiveness of the TaP attack is device type-specific, albeit for different reasons.

Vanhoef *et al.* and Schulz *et al.* [26, 29] implement targeted Wi-Fi reactive jamming on commercial devices. A reactive jammer must constantly listen to the channel, while a TaP attacker does not. Furthermore, a TaP attacker can impact several victims at once, while a reactive jammer must target each victim individually. Finally, a TaP attacker can transmit at much lower power than a reactive jammer, as explained in Section 2.1.

Previous work by Xin *et al.* introduces a physical layer testbed for benchmarking and fingerprinting IEEE 802.11 devices [32]. We use a similar experimental set-up to evaluate TaP attacks on IEEE 802.15.4 and IEEE 802.11 devices. In contrast to the setup of [32], however, the target devices are stand-alone. In particular, there is no RF cable connecting the attacking device to the target.

3 BACKGROUND

This section introduces the Truncate-after-Preamble (TaP) attack, describes the adversarial model, and introduces the metrics that will be used in subsequent sections.

3.1 Receiver State Machine

Both IEEE 802.11 and IEEE 802.15.4 are based on carrier-sense multiple access with collision avoidance (CSMA/CA). Accordingly, receivers rely on sensing the channel and detecting a preamble in order to successfully demodulate the subsequent PHY payload data (see Figure 1a).

In 802.11, the PHY receiver state machine first detects and checks all the necessary information contained in the preamble to set up reception of a payload (PSDU), and then switches into the "RX Symbol" routine which decodes individual symbols. Crucially, when the carrier is lost and subsequently no symbol can be decoded successfully, the state machine waits in the "Decrement Time" state for the "intended end of PSDU" to expire before returning back to the "RX IDLE" state in which it can detect another preamble. This behavior affects both the legacy receive PHY [12, pp. 2315-2318] as well as high throughput (HT) PHY receive procedure [12, pp. 2415-2418] equivalently.

The IEEE 802.15.4 standard does not prescribe the receiver state machine in the same level of detail [11]. This further motivates us to investigate the susceptibility of IEEE 802.15.4 devices to TaP attacks.

3.2 The Truncate-after-Preamble (TaP) Attack

The Truncate-after-Preamble (TaP) attack transmits the beginning of a PHY protocol data unit (PPDU), which includes the preamble and signal field containing payload length information, but does not transmit the actual MAC layer payload. We refer to this kind of transmitted signal as a *truncated packet.*

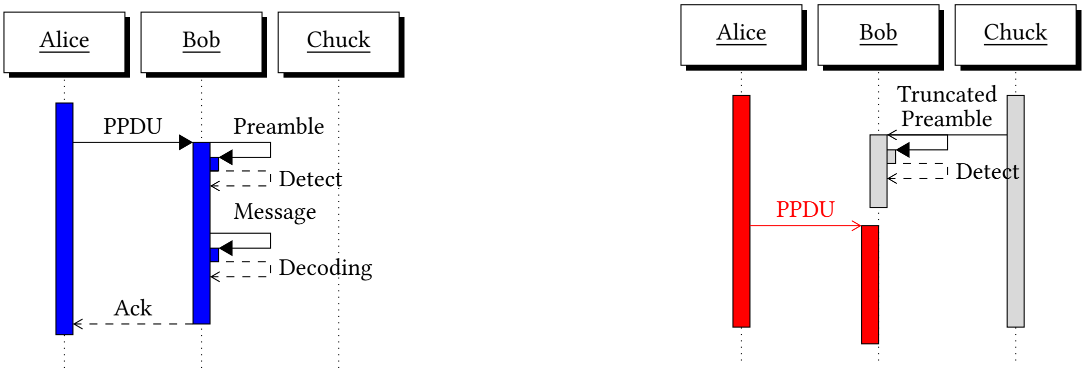
The TaP attack exploits the receiver state machine's behavior of waiting for the announced end of a transmission when failing to decode payload symbols, as illustrated in Figure 1: During normal operation (Figure 1a), Bob decodes the preamble sent by Alice (i.e., detect carrier frequency and phase, and perform frame synchronization), and proceeds to receive and decode the PSDU. If the adversary Chuck sends a preamble announcing a long PSDU, but truncates transmission before sending any payload, Bob will have successfully decoded Chuck's preamble and switched into the symbol receiving state in expectation of the announced PSDU, but fail to decode any valid symbols due to truncated transmission. According to the IEEE 802.11 protocol specification, Bob will wait for the intended end of the PSDU before returning to the "RX IDLE" state, thereby missing any incoming transmission by Alice during that time period. If Chuck sends truncated packets frequently enough as to keep Bob in the "Decrement Time" state most of the time, this can result in significant starvation of channel throughput.

We note that this is the behavior as prescribed in the protocol specification, but previous works indicate that some IEEE 802.11 devices diverge from the specified behavior [13], which makes them more or less vulnerable to the attack. We explore this issue in greater depth in Section 5.

Thus, the TaP attacker only transmits the first few raw bytes of a valid physical layer transmission and truncates these bytes before sending the bulk of the announced data. As a result, TaP is a low energy and relatively stealthy attack. A device under attack will experience failed or corrupted received packets, while the traffic visible to higher layers on the network stack will not appear congested. Note that depending on the device type and driver used, a network interface card may not report the presence of any incoming packets at all or report RX errors on the MAC layer due to incomplete packet reception.

3.3 Adversarial Model

We consider an adversary that is capable of manipulating transmissions on the physical layer. Thus, the adversary has access to specialized hardware, such as a USRP B200 software radio by Ettus Research [6] or development boards like EFR32MG1 WSTK [15] that can manipulate the low layers of the network stack in software. We assume that the channel is vacant enough for adversary's



(a) In the benign scenario, Alice sends Bob a message. Bob first detects a preamble and then successfully decodes the message. The subroutine decoding the message depends on the preamble detection subroutine completing successfully, because the preamble announces the packet length.

(b) Chuck sends a "truncated packet" announcing a payload of a certain length, but stops transmission before sending the payload (TaP attack). Bob's receiver successfully detects Chuck's preamble, expecting a message to follow, and is not in the right state to detect Alice's message despite an unobstructed channel.

Figure 1: UML Sequence Diagrams of a benign reception scenario (left) and suppressed reception via TaP attack (right). The vertical bars represent transmit processes of Alice and Chuck, as well as the receive process of Bob, respectively.

short truncated packets (512 µs for Zigbee and 20 µs for Wi-Fi) not to collide with other wireless traffic. We assume that the victim is within the transmission range of the attacker (note that the attacker could be located at significant distance from the victim if it uses a directional antenna).

As shown below, the adversary *does not* have to overpower other signals in order to launch a successful attack. In fact, the attack has low energy footprint as compared to jamming-based DoS attacks, since it relies on sending only a small fragment of a regular, protocol-compliant packet at low power.

3.4 Metrics

In this section, we present metrics that allow a characterization of the TaP attack and its effects. We first introduce a *non-experimental metric*, namely the Data Amplification Factor (DAF), that measures the potency of the attack based on protocol specifications. Next, we propose two *experimental metrics* that we evaluate for each type of individual device, namely the Preamble Receiver Starvation (PRS) and Energy Amplification Ratio (EAR) metrics.

3.4.1 Data Amplification Factor (DAF). An attack that announces data that would occupy the channel for a certain period of time while only sending for a fraction of that time is inherently more stealthy than brute-force jamming the channel. We refer to the ratio of the duration of data announced by the adversary to the duration of data actually sent as the *Data Amplification Factor (DAF)*.

Since the *DAF* only depends on how much data the attacker sends in relation to how much it pretends to send, it can be calculated directly from the parameters of each given protocol. Different attack parameters will result in different *DAF*, but even more so, different protocols lead to different maximum values.

In *Zigbee*, the TaP attack sends a total of 16 bytes at a bitrate of 250 kbit/s. Using the maximum packet length of 127 bytes results in $DAF(\text{Zigbee}) = 8.3125$.

In *Wi-Fi*, the TaP attack consists of a truncated packet of 20 µs duration (for the preamble) for which we consider an announced duration of up to 1.928 ms, resulting in $DAF(\text{Wi-Fi}) = 96.4$. The value of 1.928 ms is obtained by considering a packet with a payload of 1400 bytes transmitted at 6 Mb/s, which is the base rate for most Wi-Fi standards. In theory, many Wi-Fi devices could receive even longer packets at lower bit rates (e.g., devices backward-compatible with IEEE 802.11a or b), but we have not tested those configurations in our experiments.

In summary, for Zigbee, a TaP attack can block the channel for a duration that is 8 times longer that the attack duration. This number scales up to at least 100 times for Wi-Fi.

3.4.2 Preamble Receiver Starvation (PRS). The goal of a preamble-based starvation attack is to starve a host from receiving desired traffic by just sending a truncated packet.

Different devices behave differently in the presence of a truncated packet: While some devices maintain a substantial packet loss until the packet duration announced in the truncated packet is elapsed, other devices return to a receiving state significantly earlier [32].

Therefore, the packet p loss probability is a function of the *delay offset* Δt between the beginning of the truncated packet transmission and the beginning of the target packet transmission. For a given offset Δt, we measure the data loss probability as follows:

$$p(\Delta t) = 1 - \frac{n_{rx}(\Delta t)}{n_{tx}(\Delta t)}, \tag{1}$$

where $n_{tx}(\Delta t)$ and $n_{rx}(\Delta t)$ are the number of transmitted and received packets.

The *PRS* metric is derived by summing up the packet loss probability p experienced at discrete delay offsets $i\Delta t$, where $i = 1, 2, \ldots, N$, and normalizing it by the number of steps N:

$$PRS(d) \quad = \quad \frac{1}{N}\sum_{i=1}^{N} p(i\Delta t), \tag{2}$$

Table 1: Tested IEEE 802.15.4 devices.

Make	FCC ID	System
Marvell 88MZ100	ZKJ-12WA19	GE Wink LED LAMP 4VE8
Marvell 88MZ100	DZO-IQHOME	Osram Lightify 73674
Microchip ATSAMR21G18A	2ACQ6-A19	CREE Lightbulb A19
SiliconLabs EFR32™	FHO-ICC-A-1	IKEA LED1732G11
SiliconLabs EFR32™	N/A	EFR32™ Mighty Gecko Wireless Starter Kit (WSTK)

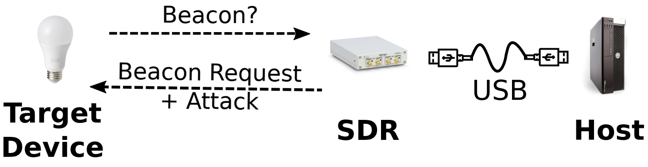


Figure 2: The over-the-air experimental setup for IEEE 802.15.4 target devices.

where d represents the device under study. In all our experiments, $N\Delta t$ corresponds to the maximum packet length duration (e.g., 512 µs for 127 bytes in Zigbee).

The PRS metric averages out the overall magnitude of receiver starvation across a range of different delay offsets. For instance, a *PRS* of 80% means that a TaP attack roughly impacts a certain device 80% as much as a brute force DoS attack that forcefully occupies the whole channel for the maximum packet duration. Note that if a device d is susceptible to continued starvation given an increased *announced* packet length, $PRS(d)$ will increase with the announced packet length.

3.4.3 Energy Amplification Ratio (EAR). Our experimental setup allows to control the gain (i.e., amplitude) of the transmitted signals. Hence, we can evaluate the effectiveness of a TaP attack for different strengths of the truncated and target packet signals. Let G_a represent the gain of the truncated packet generated by the attacker and G_t represent the gain of the target packet, which is set constant to 1. We define the *Energy Amplification ratio (EAR)* of a device d as

$$EAR(d) = \max_{G_a} 20 \log_{10}\left(\frac{G_t}{G_a}\right), \tag{3}$$

such that the attack succeeds with high probability (say above 90%). This metric measures how weak the adversarial truncated packet can be relative to the legitimate traffic (i.e., how small G_a can be), with the attack still being effective. This metric typically depends on individual device characteristics, such as dynamic range and sensitivity.

4 ZIGBEE

4.1 Experimental Setup

The experimental setup for testing IEEE 802.15.4 devices resembles the IEEE 802.11 physical layer testbed introduced [32] but is based on the gr-ieee-802-15-4 transceiver [3] for generating and receiving Zigbee packets, as well as custom blocks to perform the PHY packet truncation.

In contrast to the setup of [32], we use stand-alone target devices (see Table 1 for the list of devices). We do not use RF cables and do not need to connect a host device to collect packet loss statistics. As shown in Figure 2, we instead broadcast IEEE 802.15.4 "beacon request" packets from the SDR. These requests are acknowledged with a "beacon" packet by the tested device (light bulbs). We measure the presence of these beacons under different adversarial conditions. This set-up is more realistic for typical IoT devices, such as smart home sensors, that operate as stand-alone devices and are not meant to be programmatically controlled by a host device during normal operations. Note that the actual TaP DoS attack only requires periodically sending truncated beacon requests (or other suitable frame types) with maximal frame length without necessarily listening for responding beacons.

4.2 Starvation Dependence on Announced Packet Length

We investigate how IEEE 802.15.4 devices are impacted by a TaP attack, specifically how long and with what probability a device appears as busy (i.e., it cannot receive a new packet) after receiving a truncated packet.

Experiment. The first experiment measures if and how *receiver starvation* (i.e., a negatively impacted reception capability) occurs as a function of (i) the frame length field inside the truncated (DoS) packet, and (ii) the delay offset between the truncated packet and the test packet.

In this experiment, the adversary announces packet lengths of 0, 32, 64, 96, or the maximum possible value of 127 bytes, but truncates transmission after 10 bytes into the payload. This is repeated for delay offsets that are multiple of Δt = 250 µs, i.e., {250 µs, 500 µs, 750 µs, ...,5000 µs}. The relative signal strengths of the adversarial truncated packet and the target packet are set to $G_a = 0.1$ and $G_t = 1.0$, respectively, i.e., the target packet one order of magnitude (20dB) more powerful than the attack. For each given parameter configuration, we display results averaged over 100 experiments and provide 95% confidence intervals.

Results. The results of the experiments are shown in Figure 3. All the five tested IEEE 802.15.4 devices are vulnerable to the TaP attack. While the adversary transmits only during the "attack window" shaded in pink, close to 100% packet loss occurs for the whole range of delay offsets (shaded in violet) which span the *announced* length – even though the channel remained physically free during all the time windows shaded in violet.

We next assess the five devices with respect to the Preamble Receiver Starvation (PRS) metric. As shown in Figure 4, all measured

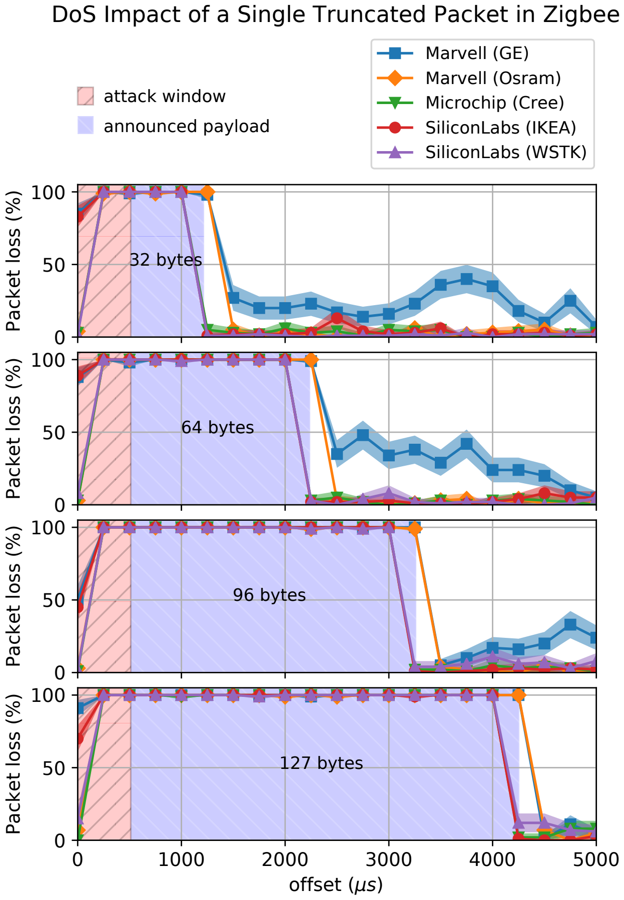
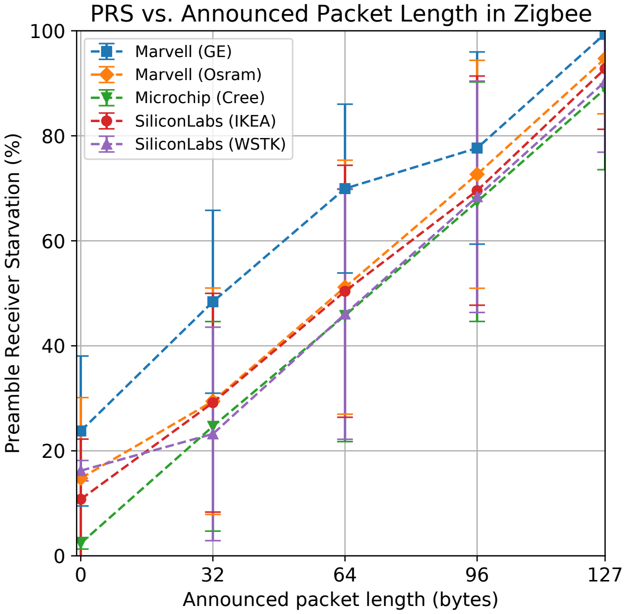


Figure 3: All five tested Zigbee IoT devices are impacted by the TaP attack, as visualized by the discrepancy between the period in which the adversary is actually transmitting (pink) and the period during which devices experience receiver starvation, i.e., substantial packet loss induced by the attack. All devices experience full packet loss until the end of the announced payload and subsequently return to their regular response behavior (with Marvell (GE) exhibiting a slightly higher baseline packet loss than the other devices). The colored bands represent 95% confidence intervals.

Zigbee devices have a mostly linear PRS slope depending on the announced packet length, which is consistent with Figure 3. Thus, the longer the announced packet length, the higher the percentage of undelivered packets. Note that the slope of the PRS function indicates how fast each chip's reception block increases with reservation times (i.e., the higher the slope, the more vulnerable the radio chip). With an announced packet length of 127 bytes, the PRS for all the chips exceed 90%, i.e., the attack is 90% as effective as brute-force jammer that transmits continuously over the same maximum packet length duration.



'

Figure 4: The Preamble Receiver Starvation (PRS) is highly impacted by the announced packet length in common Zigbee chipsets. All the devices exhibit similar behavior, as shown by the largely overlapping 95% confidence intervals.

4.3 Starvation Dependence on Attacker's Signal Strength

We next assess the minimal power an attacker has to invest to perform an effective attack, using the Energy Amplification Ration (EAR) metric.

Experiment. In this experiment, we keep the announced packet length fixed at a value of 127 bytes, and instead vary the adversary's signal gain G_a. A variation from a factor of 10^{-4} to 1 relative to the signal amplitude results in the range of 80 dB to 0 dB EAR, which we measure in steps of 5 dB.

After running the delay test at all of these values for all devices, we compute the *EAR* metric by selecting the lowest gain G_a at which an attack still occurs with at least 90% effectiveness.

Results. As shown in Figure 5, for all Zigbee devices, an attack with EAR up to 50 dB is still effective (i.e., $G_a \approx 3 \cdot 10^{-3}$), and up to 60 dB for some devices (i.e., $G_a \approx 10^-3$). Thus, a TaP attack can succeed even when the adversary's signal strength is 300 times weaker than the signal strength of the victim's traffic it is impacting. Combining this insight with the Data Amplication Factor (FAF) of 8.3125 for Zigbee, an attacker can launch an effective attack that spends about six orders of magnitude less transmit energy than that of a constant jammer (assuming the constant jammer's power is roughly the same as the target packet's power).

Table 2: Tested IEEE 802.11 cards.

Make	Model	Interface	Protocols	Chipset
AmazonBasics	Wi-Fi 11N USB Adapter - 300 Mbps	USB	b/g/n	Realtek RTL8192EU
D-Link	DWL-G122 rev B1	USB	b/g	Ralink RT2570
Intel	AX200	m.2	b/g/n/ac/ax	Intel AX200NGW
SparkLAN Communications	WUBM-273ACN	USB	b/g/n/ac	Mediatek MT7612UN
Panda Wireless	PAU06 300Mbps N	USB	b/g/n	Ralink RT5372
TP-Link	TL-WN722N N150	USB	b/g/n	Atheros AR9271
TP-Link	TL-WN822N (Dual Antenna)	USB	b/g/n	Realtek RTL8192EU

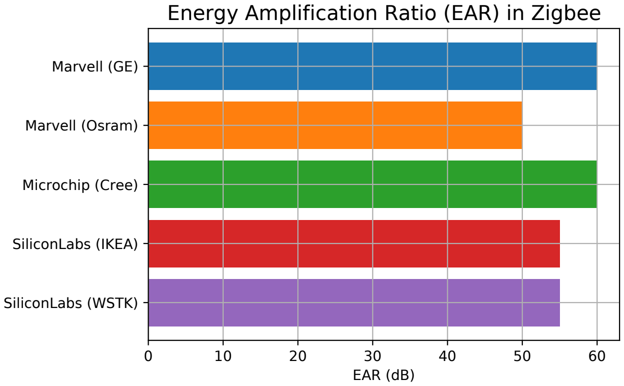


Figure 5: In IEEE 802.15.4, the TaP attack succeeds even when the attacker's signal strength is 50 dB weaker than the target signal's strength.

5 WI-FI

5.1 Experimental Setup

The experimental setup for testing IEEE 802.11 relies on a physical layer testbed that resembles that of [32], but with the key difference of transmitting the generated signal over-the-air rather than via RF cables. This allows testing of devices that do not have an easily accessible antenna port, such as the TPLink WN822N. Table 2 provides the list of tested Wi-Fi devices.

We control the host machines of both the transmitter and the receiver. On the transmission side, we use the gr-ieee-802-11 library for GNU Radio [4] to generate probe request frames at precise relative delay offsets relative to the truncated packets which consist of a preamble with configurable length field, but without any PHY payload. On the reception side, we log received packets via `tcpdump` on the host of the tested device and calculate packet drop statistics based on the quantity of sent vs. received packets. The source code used to generate truncated packets is provided [18].

5.2 Starvation Dependence on Announced Packet Length

We investigate how long and with what probability a device appears as busy after receiving a truncated packet.

Experiment. In this experiment, the adversary announces packet lengths of 200, 400, 600, 800, 1000, 1200, and 1400 bytes, but only send a 20 µs preamble. This is repeated 100 times for each of these settings, and for delay offsets that are integer multiple of Δt =10 µs. The relative signal strengths of the adversarial truncated packet and the target packet are set to $G_a = 0.1$ and $G_t = 1.0$, respectively, i.e., the target packet one order of magnitude (20 dB) more powerful than the attack.

Results. Figure 6 shows the performance of the seven Wi-Fi devices with respect to the Preamble Receiver Starvation (PRS) metric. We observe that the devices widely differ in terms of their vulnerability to TaP attacks. Some devices, such as the Panda PAU06, are widely susceptible to the attack as the PRS keeps increasing almost linearly with the announced packet length, reaching a value as high as 80%. Yet, for other cards, such as the TPLink WN722N, DLink DWL-G122, and Intel AX200, the PRS curve remains almost flat somewhere between 5% and 10%. While these devices are not entirely immune to the TaP attack, their starvation does not worsen when increasing the announced packet length. The reason is that the receiver state machine of these devices do not wait until the end of the announced packet length in order to start receiving new packets.

5.3 Starvation Dependence on Attacker's Signal Strength

We next assess the performance of the Wi-Fi devices in terms of the EAR metric.

Experiment. In this experiment, we keep the announced packet length fixed at a value of 200 bytes. We vary the adversary's signal gain G_a from 10^{-4} to 1 leading to an EAR respectively ranging from 80 dB to 0 dB EAR, in steps of 5 dB.

After running the delay test at all of these values for all devices, we compute the *EAR* metric by selecting the lowest gain G_a at which an attack still occurs with at least 90% effectiveness.

Results. As shown in Figure 7, for all Wi-Fi devices, the attack succeeds with an EAR of up to 30 dB (i.e., $G_a \approx 3 \cdot 10^{-2}$), and even a higher EAR for some devices. Thus, the Wi-Fi TaP attack succeeds even when the adversary's signal strength is about 30 times weaker than the signal strength of the victim's traffic it is impacting. Combining this insight with the Data Amplication Factor (DAF) of 96.4 for Wi-Fi, an attacker can launch an effective attack that spends about five orders of magnitude less transmit energy than that of a constant jammer.

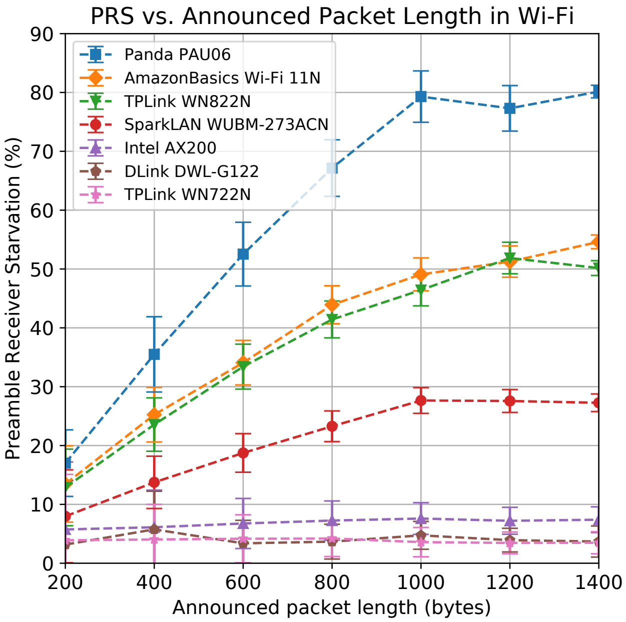


Figure 6: Preamble Receiver Starvation (PRS) performance of different Wi-Fi devices. The devices are not equally susceptible to the TaP attack.

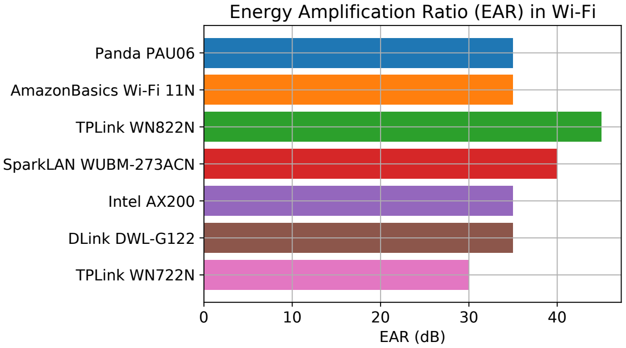


Figure 7: In IEEE 802.11, the TaP attack succeeds even when the attacker's signal strength is 30 dB weaker than the target signal's strength.

6 DETECTION METHODS

We next discuss three methods to help detect an ongoing preamble-based starvation attack. These method require different levels of equipment to perform.

NIC-level detection. The first level of detection can be achieved with a regular NIC; however, it can only provide qualitative confirmation of a necessary symptoms of an ongoing preamble starvation attack, not a definite proof of its presence. The effects of a preamble-based starvation attack on a NIC are characterized by the presence of two observations on a given channel that are usually not made at the same time:

- The channel throughput achieved by any device suggests a heavily congested channel (many RX errors or CRC failures).
- However, there is barely any traffic visible on the channel (few successful RX-es).

While these circumstances can easily be identified with regular wireless devices and network scanning tools such as Wireshark (Figure 8), these symptoms may also be caused by other physical-layer interference that degrades medium access. The absence of other plausible interference sources may raise suspicion and could be followed up with one of the more elaborate detection methods below.

Time-domain detection. The second level of detection requires measuring the RF signal on a capable oscilloscope or with an SDR. Figure 9 shows the raw signal of a truncated packet in relation to a typical Wi-Fi frame to illustrate the proportions in time and energy to each other. The measurement was performed on an ADALM-PLUTO SDR [1], using the Universal Radio Hacker software [21]. The truncated packet is clearly identifiable on the picture.

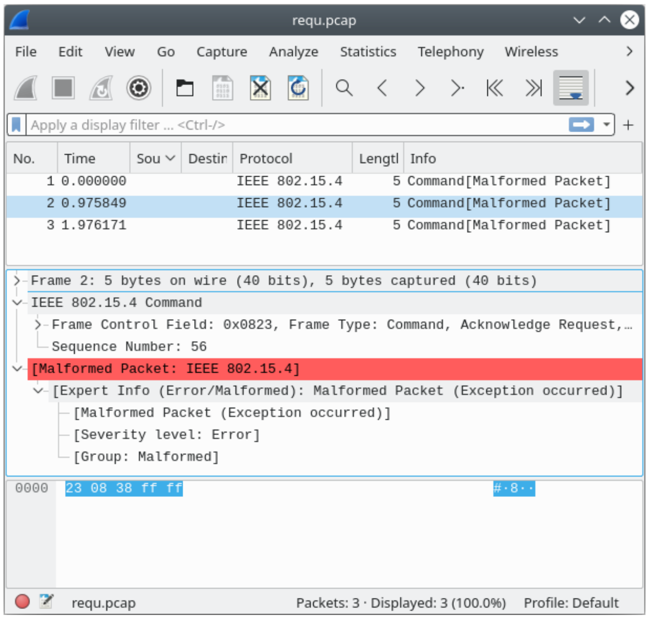
Note, however, that in this example the signal strength of the truncated packet is the same as that of the target packet. As discussed earlier, the attack may still be effective at much lower signal strength, making it harder to detect. Also, short burst of traffic may be caused by legitimate short packets in Wi-Fi (e.g., ACKs, RTS, CTS) or other non-Wi-Fi protocols, possibly leading to false positives.

Preamble-Counter detection. The third level of detection requires direct access to the Data-Link/PHY layer, i.e., a network stack implementation that allows granular inspection into the raw physical-layer incoming data frames as they are demodulated. This can be achieved in real time with a RAIL Test Application for EFR32 development board [15] through the packet trace interface (PTI) debugging pins. The received signals can be recorded with a logic analyzer (see Figure 10). Using such a setup, it is possible to collect the occurrence of successful preamble detection as well as successful frame decoding. A high number of detected preambles compared to a lower number of successfully decoded frames strongly indicates the occurrence of a TaP attack. In addition, one can count the number of aborted packets due to packet length error and use this information as another indicator of a TaP attack.

7 CONCLUSION

We designed and demonstrated the feasibility of Truncate-after-Preamble (TaP) attacks on IEEE 802.15.4 and IEEE 802.11 devices, which represent a large class of IoT devices currently deployed. We showed that all tested Zigbee devices are vulnerable to the attack, independently of the manufacturer. Wi-Fi devices are vulnerable to the attack to varying degrees. The attacks are five to six orders of magnitudes more energy-efficient than constant jamming attacks, because the attacker transmits only a small fraction of the time and the signal strength can be much lower than the target signal strength.

Specifically, we showed that a truncated 802.15.4 packet lasting for 512 µs causes a denial-of-service lasting around 4.3 ms with an attacker signal strength at least 300 times weaker than the target signal strength, for all cards tested. Similarly, a truncated IEEE 802.11



```
No.  Time      Sou  Destir  Protocol       Lengtl  Info
   1 0.000000           IEEE 802.15.4       5 Command[Malformed Packet]
   2 0.975849           IEEE 802.15.4       5 Command[Malformed Packet]
   3 1.976171           IEEE 802.15.4       5 Command[Malformed Packet]

> Frame 2: 5 bytes on wire (40 bits), 5 bytes captured (40 bits)
v IEEE 802.15.4 Command
   > Frame Control Field: 0x0823, Frame Type: Command, Acknowledge Request,…
     Sequence Number: 56
v [Malformed Packet: IEEE 802.15.4]
   v [Expert Info (Error/Malformed): Malformed Packet (Exception occurred)]
        [Malformed Packet (Exception occurred)]
        [Severity level: Error]
        [Group: Malformed]

0000  23 08 38 ff ff                                    #·8··
```

Figure 8: A truncated/malformed packets detected in Wireshark using USRP SDR .

packet lasting for 20 µs can cause up to 1.92 ms denial-of-service (in the worst-case) with a signal strength 30 times weaker. Wi-Fi has a substantially greater DAF (data amplification factor) due to the larger packet sizes relative to the preamble. Yet, attacks on Zigbee have a higher Energy Amplification Ratio (EAR) reaching 50 dB and above, while the highest EAR effective against all the Wi-Fi cards is 30 dB.

Interestingly, one cannot strictly correlate the TaP vulnerability to different chipset vendors. For instance, in Figure 6, the Panda Wireless and D-Link cards both use Ralink chipsets but have significantly different responses to the attack, while the Amazon Basics and TPLink WN822N cards both use the RTL8192EU chipset and have very similar responses. We conclude that susceptibility to the TaP attack highly depends on the firmware. Ultimately, such configurations trade off between security, performance, and in the case of Wi-Fi, adherence to the protocol specification.

7.1 Future Work

The TaP attack may deny the availability of various kinds of mission-critical devices, such as IoT medical devices [22], industrial production sensors [28], public infrastructure sensors [14], and fire alarms in smart buildings [10] for an extended period of time.

The energy efficiency of the TaP attack could allow for stealthy, mobile deployment of the attack. For instance, an EFR32 radio transmitting at 20dBm consumes about 120mA [16]. A malicious device carrying out the attack via Zigbee requiring about 12.5% duty cycle for complete DoS could last up to about 16 hours on a coin cell battery[1].

While our paper demonstrated the effectiveness of TaP attacks against Zigbee devices, many other IoT protocols such as WirelessHART, 6LowPAN, Thread, etc. are also based on the IEEE 802.15.4 standard. As part of future work, it would be useful to assess whether devices using such protocols are indeed vulnerable to TaP attacks.

We expect that TaP attacks could affect numerous products based on these protocols, such as Amazon Echo, Nest Thermostat, Samsung SmartThings Gateway [9] and medical devices, such as insulin pumps and CGM (continuous glucose meter) devices [22] which are running non-standard proprietary low-bit rate IoT protocols. Moreover, these many protocols are running on a large number of devices which all tend to be based on only a few popular chip vendors [27], which could amplify the attack surface to the whole IoT ecosystem. In future work, we plan to investigate how the TaP physical layer attack could ultimately affect the application layer of the mentioned products.

ACKNOWLEDGMENTS

This research was supported in part by the U.S. National Science Foundations under grants CNS-1409453 and CNS-1908087. The authors also thank Dr. Liangxiao Xin for fruitful discussions.

REFERENCES

[1] Analog Devices. 2017. ADALM-PLUTO | Software-Defined Radio Active Learning Module. https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/adalm-pluto.html

[2] John Bellardo and Stefan Savage. 2003. 802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions.. In *USENIX security symposium*, Vol. 12. Washington DC, 2–2.

[3] Bastian Bloessl, Christoph Leitner, Falko Dressler, and Christoph Sommer. 2013. A GNU Radio-based IEEE 802.15.4 Testbed. In *12. GI/ITG KuVS Fachgespräch Drahtlose Sensornetze (FGSN 2013)*. Cottbus, Germany, 37–40.

[4] Bastian Bloessl, Michele Segata, Christoph Sommer, and Falko Dressler. 2018. Performance Assessment of IEEE 802.11p with an Open Source SDR-based Prototype. *IEEE Transactions on Mobile Computing* 17, 5 (may 2018), 1162–1175. https://doi.org/10.1109/TMC.2017.2751474

[5] David H. Deans. 2019. Wi-Fi device shipments will hit four billion by 2024: The ramifications for the industry. https://www.telecomstechnews.com/news/2019/jul/16/wi-fi-device-shipments-will-reach-4-billion-by-2024/

[6] Ettus Research. 2019. USRP B200. https://kb.ettus.com/B200/B210/B200mini/B205mini

[7] Wenjun Gu, Zhimin Yang, Dong Xuan, Weijia Jia, and Can Que. 2009. Null data frame: A double-edged sword in IEEE 802.11 WLANs. *IEEE Transactions on Parallel and Distributed Systems* 21, 7 (2009), 897–910.

[8] Joseph Hall, Benjamin Ramsey, Mason Rice, and Timothy Lacey. 2016. Z-Wave Network Reconnaissance and Transceiver Fingerprinting Using Software-Defined Radios. , 163-X pages.

[9] Matt Hamblen. 2019. Silicon Labs emerging as rising heavyweight in IoT electronics. https://www.fierceelectronics.com/electronics/silicon-labs-emerging-as-rising-heavyweight-iot-electronics

[10] Yong Yang Huiyu Wu, Yuxiang Li. 2018. Hacking Intelligent Buildings: Pwning KNX & ZigBee Networks. https://conference.hitb.org/hitbsecconf2018ams/sessions/hacking-intelligent-buildings-pwning-knx-zigbee-networks/

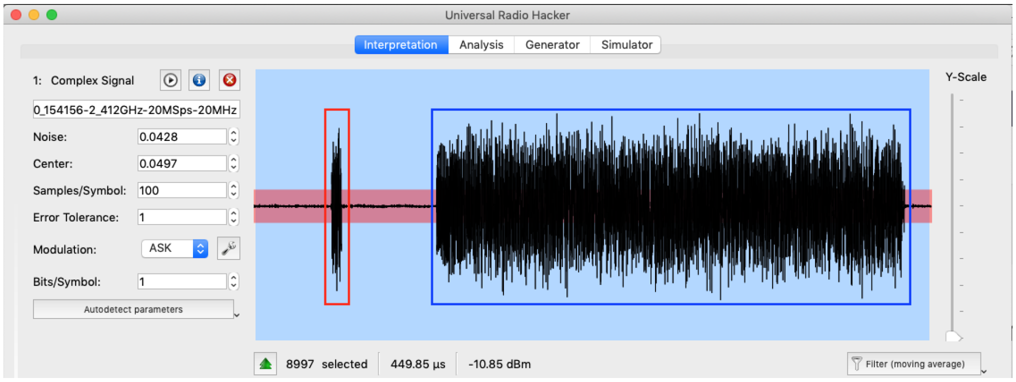
[11] IEEE Standards Association. 2015. *802.15.4-2015 - IEEE Standard for Low-Rate Wireless Networks*. IEEE, New York, New York, USA.

[12] IEEE Standards Association. 2016. *802.11-2016 - IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Sp*. IEEE. https://doi.org/10.1109/IEEESTD.2016.7786995

[13] Evgeny Khorov, Aleksey Kureev, Ilya Levitsky, and Andrey Lyakhov. 2018. Testbed to Study the Capture Effect: Can We Rely on this Effect in Modern Wi-Fi Networks. In *2018 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*. IEEE, 1–5. https://doi.org/10.1109/BlackSeaCom.2018.8433688

[14] Irina Krivtsova, Ilya Lebedev, Mikhail Sukhoparov, Nurzhan Bazhayev, Igor Zikratov, Aleksandr Ometov, Sergey Andreev, Pavel Masek, Radek Fujdiak, and

[1] Assuming a coin cell battery capacity of 240mAh, an EFR32 radio would last up to 240mAh/(0.125 · 120mA) = 16h, ignoring other energy consumption on the battery cell.



0_154156-2_412GHz-20MSps-20MHz

Noise: 0.0428

Center: 0.0497

Samples/Symbol: 100

Error Tolerance: 1

Modulation: ASK

Bits/Symbol: 1

8997 selected 449.85 µs -10.85 dBm

Figure 9: A signal recorded with urh [21] on an ADALM-PLUTO [1] shows the difference between a truncated preamble (left, delineated in red) and an actual packet (right, delineated in blue). The x-axis represents time, and the total duration of the signal captured here is 450 µs. When a legitimate packet follows a truncated preamble with such a short delay as shown in this scenario (≈70 µs), a vulnerable device might not receive the message delineated in blue despite the absence of any collisions.

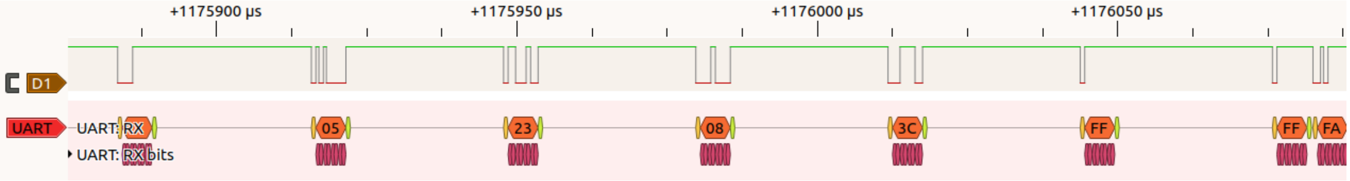


Figure 10: A truncated 802.15.4 beacon request can be detected using a logic analyzer through PTI (packet trace interface) debug pins of a EFR32MG1 development board [15]. The top trace shows the received frame bytes plus metadata bytes as a raw logic analyzer signal. The bottom trace shows the decoded bytes in hexadecimal format. The first byte of 0x05 is the fake packet length. Next, the bytes 0x23 and 0x08 (0x0823) correspond to the frame control field. The byte 0x3C represents the sequence number. The bytes 0xFFFF are the destination PAN (personal area network) address. The final byte 0xFA means that the packet is aborted, in this case due to the incorrect packet length.

Jiri Hosek. 2016. Implementing a broadcast storm attack on a mission-critical wireless sensor network. In *International Conference on Wired/Wireless Internet Communication*. Springer, 297–308.

[15] Silicon Labs. 2010. EFR32™ Mighty Gecko Wireless Starter Kit. https://www.silabs.com/products/development-tools/wireless/mesh-networking/mighty-gecko-starter-kit

[16] Silicon Labs. 2016. EFR32MG1 Mighty Gecko Multi-Protocol SoC Family Data Sheet. https://www.silabs.com/documents/public/data-sheets/efr32mg1-datasheet.pdf

[17] Rohit Negi and Arjunan Rajeswaran. 2005. DoS analysis of reservation based MAC protocols. In *IEEE International Conference on Communications, 2005. ICC 2005. 2005*, Vol. 5. IEEE, 3632–3636.

[18] NISLAB @ Boston University. 2020. Github Repositories. https://github.com/nislab

[19] Colin P O'Flynn. 2011. Message denial and alteration on IEEE 802.15. 4 low-power radio networks. In *2011 4th IFIP International Conference on New Technologies, Mobility and Security*. IEEE, 1–5.

[20] Konstantinos Pelechrinis, Marios Iliofotou, and Srikanth V Krishnamurthy. 2010. Denial of service attacks in wireless networks: The case of jammers. *IEEE Communications surveys & tutorials* 13, 2 (2010), 245–257.

[21] Johannes Pohl and Andreas Noack. 2018. Universal Radio Hacker: A Suite for Analyzing and Attacking Stateful Wireless Protocols. In *12th USENIX Workshop on Offensive Technologies (WOOT 18)*. USENIX Association, Baltimore, MD. https://www.usenix.org/conference/woot18/presentation/pohl

[22] Jerome Radcliffe. 2011. Hacking medical devices for fun and insulin: Breaking the human SCADA system. In *Black Hat Conference presentation slides*, Vol. 2011.

[23] Hanif Rahbari, Marwan Krunz, and Loukas Lazos. 2015. Swift jamming attack on frequency offset estimation: The Achilles' heel of OFDM systems. *IEEE Transactions on Mobile Computing* 15, 5 (2015), 1264–1278.

[24] B. Ramsey, J. Fuller, and C. Badenhop. 2016. Efficacy of physical layer preamble manipulation for IEEE 802.11a/ac. *Electronics Letters* 52, 8 (2016), 669–671. https://doi.org/10.1049/el.2015.4228

[25] B. W. Ramsey, B. E. Mullins, M. A. Temple, and M. R. Grimaila. 2015. Wireless Intrusion Detection and Device Fingerprinting through Preamble Manipulation. *IEEE Transactions on Dependable and Secure Computing* 12, 5 (Sep. 2015), 585–596. https://doi.org/10.1109/TDSC.2014.2366455

[26] Matthias Schulz, Francesco Gringoli, Daniel Steinmetzer, Michael Koch, and Matthias Hollick. 2017. Massive reactive smartphone-based jamming using arbitrary waveforms and adaptive power control. In *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. 111–121.

[27] Technavio. 2016. Top 9 Vendors in the ZigBee Home Automation Market from 2016 to 2020: Technavio. *Business Wire* (2016). https://www.businesswire.com/news/home/20160906005343/en/Top-9-Vendors-ZigBee-Home-Automation-Market

[28] Ann R. Thryft. 2018. Real-Life Industrial IoT Cyberattack Scenarios. https://www.eetimes.com/real-life-industrial-iot-cyberattack-scenarios/

[29] Mathy Vanhoef and Frank Piessens. 2014. Advanced Wi-Fi attacks using commodity hardware. In *Proceedings of the 30th Annual Computer Security Applications Conference*. 256–265.

[30] Matthias Wilhelm, Ivan Martinovic, Jens B Schmitt, and Vincent Lenders. 2011. Short paper: reactive jamming in wireless networks: how realistic is the threat?. In *Proceedings of the fourth ACM conference on Wireless network security*. 47–52.

[31] ON World. 2019. 802.15.4 IoT Markets. https://onworld.com/research/zigbee/vip/

[32] Liangxiao Xin, Johannes K. Becker, Stefan Gvozdenovic, and David Starobinski. 2019. Benchmarking the Physical Layer of Wireless Cards Using Software-Defined Radios. In *Proceedings of the 22nd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM '19)*. Association for Computing Machinery, New York, NY, USA, 271–278. https://doi.org/10.1145/3345768.3355907

[33] Wenyuan Xu, Ke Ma, Wade Trappe, and Yanyong Zhang. 2006. Jamming sensor networks: attack and defense strategies. *IEEE network* 20, 3 (2006), 41–47.