Sensorless, Permissionless Information Exfiltration with Wi-Fi Micro-Jamming

Rom Ogen, Omer Shwartz, Kfir Zvi, Yossi Oren
Ben-Gurion University of the Negev
romog@post.bgu.ac.il, omershv@post.bgu.ac.il, zvikf@post.bgu.ac.il, yos@bgu.ac.il

Abstract

Listening devices, tracking devices, and other covert implants have to send any data they collect to a central command and control (C&C) server. This task can be difficult, since implants typically have a restricted power budget and cannot connect directly to the Internet. Several works have attempted to exfiltrate data in this setting by taking advantage of a nearby networked device, such as a computer or a mobile phone. To achieve this, the implant uses a covert channel to send the data to the networked device, that performs the exfiltration. Several constructions have been proposed for this covert channel between implant and target device, using sensors such as the microphone, magnetometer and gyroscope.

In this work, we implement this covert channel using Wi-Fi micro-jamming, a new approach to jamming the 802.11 Wi-Fi protocol in a low-power, minimally intrusive manner. Our construction, which extends the work of Shah and Blaze from WOOT '09 [1], does not attempt to overwhelm the Wi-Fi channel with a high-power transmission, but instead takes advantage of the high sensitivity of the 802.11 protocol's Clear Channel Assessment (CCA) mechanism to introduce very brief delays to the channel. A JavaScript program, which can be embedded in an attacker-controlled website or online advertisement, is then used to measure these delays and upload them to the C&C server.

Our channel works at a distance of over 15 meters between implant and target device, achieves a bit rate of 40 bits per second with minimal errors, and has a very low power requirement. We even show how the implant can be made completely passive by replacing the transmit antenna with a backscattering antenna, making its location very hard to detect. Most importantly, since our attack uses only Wi-Fi communications, it works on a wide variety of devices with different form factors and requires no extra permissions on the receiver's side. This makes it very difficult to defend against this attack using existing information flow control countermeasures.

1 Introduction

Humanity has been worrying about espionage and eavesdropping ever since the invention of the first secret [2]. Traditional eavesdropping required an agent to be in the vicinity of the target. As technology has developed, it brought with it a vast increase in the range and capabilities of eavesdropping. Even before the advent of digital recording, analog surveillance bugs, capable of capturing and transmitting information, were used by rival countries [3]. Today, a plethora of monitoring and privacy-intruding devices exist for well-funded organizations as well as hobbyist spies.

As noted by Farshteindiker et al. [4], a modern surveillance implant usually consists of three logical components: a **sensor**, a **power source** and an **exfiltration mechanism**. The sensor performs data acquisition, listening for speech, movements, keystrokes or other actions of the target. The power source provides the device with power for its computation and communication functions. Examples of power sources include batteries, external power connections or even passive power harvesting. Finally, the exfiltration mechanism provides the means to transmit recorded secrets to the attacker's C&C server.

In the rest of this paper, we assume that the secret information has been already collected and deal only with the issue of exfiltration. Exfiltration can be easy when unregulated wired or wireless Internet access is available. Implants, however, operate in an adversarial setting, meaning that they cannot simply authenticate and connect to their victim's network. Implants rarely contain a cellular modem, satellite radio or other types of long-range radio transmitter either, since these functions are all power-hungry and easily detectable. It is also problematic to use short-range radio communications to perform the exfiltration, since this requires the implant's owner to send a field agent, equipped with a sophisticated collection device, to the vicinity of the intelligence target

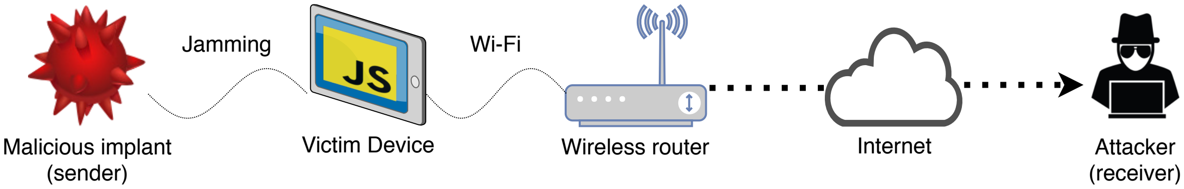


Figure 1: The general attack model

[5], an endeavor which is both costly and risky.

Several works have attempted to exfiltrate data in this setting by having the implant take advantage of a nearby networked device, such as a computer or a mobile phone. To achieve this, the implant first uses a covert channel to send the data to the networked device. Next, some code running on the networked device makes it act as the receiver, causing the device to recover the secret data and then perform the actual act of exfiltration. Several constructions have been proposed for this covert channel between implant and target device, mostly using sensors such as the microphone, magnetometer and gyroscope. The main limitation of many of these sensor-based approaches is that a properly-implemented information flow control policy can prevent untrusted applications from accessing these sensors, effectively requiring the implant owner to completely compromise a victim device before the implant can be used.

1.1 Our contributions

In this work, we present a new way for implants to exfiltrate information, based on jamming the Wi-Fi shared medium for extremely short periods of time. We call our approach micro-jamming. Our construction extends the initial work of Shah and Blaze from WOOT '09 [1], who proposed this "interference channel" mechanism and evaluated it using a high-powered radio platform. Two key advantages of our approach are the fact that its power requirements are extremely modest, and the fact that it asks nothing more from the victim device than to run some unprivileged JavaScript code. The general structure of our system is illustrated in Fig. 1. As shown in the figure, the malicious implant sends its secret onto the Wi-Fi shared medium in the form of short jamming sequences. Unprivileged JavaScript code running on the target networked device probes the network to detect the presence of these jamming sequences, then uploads its measurements to the attacker's C&C server.

More specifically, our paper makes the following contributions:

1. We describe several features of the Wi-Fi protocol which make an interference-based covert channel over Wi-Fi highly viable and robust.
2. We implement a Wi-Fi covert channel which uses a low-powered active hardware implant as the sender and a permission-less JavaScript file as the receiver.
3. We explore the boundaries and parameters of our system and show how it overcomes several significant limitations seen in prior works. Some of the capabilities of the system are:
   (a) Reliable data transfer at up to 40 bits per second.
   (b) Viability in ranges of over 15 meters.
   (c) Effectiveness in transmission powers as low as -17 dBm, or 20 microwatts.
   (d) Has only a minor impact on the throughput of the network used for exfiltration, with a natural trade-off between duty cycle and bit error rate.
   (e) Depends on no sensors, permissions or installed software on the target device
4. We present a variant of our implant which incorporates techniques from the world of RFID, allowing exfiltration to be performed using passive powerless modulation of existing signals. This design, which is a simplified variant of Passive Wi-Fi [6], dramatically reduces the power requirements of the implant and makes it very difficult to detect.

Document Structure. The rest of the paper is organized as follows. In Section 2 we describe the carrier sense mechanism of the 802.11 Wi-Fi protocol and its sensitivity to jamming. Next, in Section 3 we design and implement an exfiltration mechanism based on Wi-Fi micro-jamming and evaluate its performance. In Section 4 we discuss how the power consumption and detectability of the implant can be improved by making it completely passive, and show some results toward achieving this goal. Finally, we conclude in Section 5 with a discussion of countermeasures for our attack and directions for future research.

1.2 Related Work

This paper shows how a low-powered implant can exfiltrate data by using properties of the Wi-Fi radio protocol. The idea of interfering with the Wi-Fi protocol to exfiltrate data was first suggested by Shah and Blaze in WOOT '09 [1]. Shah and Blaze introduced the concept of an "interference channel", which they defined as a "covert channel that works by creating external interference on a shared communications medium (such as a wireless network)". The advantage of the interference channel over traditional overt channels is that the covert sender does not need to compromise a host, or to authenticate itself to the shared medium, in order to communicate. Shah and Blaze produced one implementation of such an interference channel, with the covert sender consisting of a high-powered Wi-Fi jammer with a peak power output of 20 dBm (100 milliwatts), and a covert receiver consisting of a pair of native code programs which exchange a constant stream of UDP network packets. Using this setup, Shah and Blaze were able to encode 0.4 bits per second of data into network traffic, while achieving a decoding accuracy of around 90%.

Our work extends the work of Shah and Blaze in several dimensions. Our first improvement is a **reduced transmit power**: instead of a high-powered jammer transmitting at 20 dBm, we present an active jammer transmitting at power levels as low as -17 dBm, signal level which is nearly four orders of magnitude lower than Shah and Blaze. We also built a **completely passive jammer** which does not transmit at all, but rather assumes a pre-existing energy source in a different channel, then takes advantage of the backscattering effect to perform the jamming operation. Our second improvement is an **increased bit-rate**: instead of 0.4 bits per second, our method achieves near-perfect decoding accuracy at a rate of 40 bits per second, enough for operating a keylogger or similar text-based exfiltration channels. Our third and most significant improvement is to the **attacker model**: to measure the effect of the sender's interference on the channel, Shah and Blaze assumed that the receiver has on-path capability, that is, it can capture the packets affected by the sender and then analyze them using a native code program. One application suggested by Shah and Blaze was the watermarking of VoIP packets, which can presumably be picked up on the other end of the conversation. In our work, in contrast, we consider a less powerful attacker who only has the ability to run unprivileged JavaScript code on a host affected by the interference channel. This model is justified in many cases, including active on-path attackers [7], malicious apps or malicious web ads, and provides us with great flexibility in the choice of hardware and software platforms for the target, including laptops and mobile phones from multiple vendors.

Several related works used other methods for covert data exfiltration, using various sensors found on phones and laptops, including the phone's magnetic compass, gyroscope, microphone, speaker and camera [8, 9, 10, 4]. For example, in WOOT '14 Deshotels presented a channel between two Android smartphones based on ultrasonics[10]. Among all of these works, the one which is the most similar to ours is the work of Farshteindiker et al. from WOOT 2016, which used vibrations of the gyroscope sensors found on mobile phones and laptops as a covert channel [4]. Similar to our work, the communication channel described by Farshteindiker et al. does not require any unprivileged code to run on the victim's phone or laptop; specifically, it can be deployed in the form of an untrusted webpage. Our main improvement over the work of Farshteindiker et al. is the increased distance between the implant and the victim: while Farshteindiker et al. required direct physical proximity between the implant and the victim's phone or laptop, our system works at ranges of over 15 meters, as we demonstrate in Section 3. An additional advantage of our work is that it does not presuppose that any type of sensor is present on the target device. While gyroscopic sensors are indeed found in all mobile phones and in several types of laptops, Wi-Fi radios are far more ubiquitous. In addition, while future web standards may make the gyroscope API available only to privileged webpages [11], it is highly unlikely that an untrusted webpage would ever be prevented from using Wi-Fi.

2 Jamming the 802.11 protocol

This section describes several characteristics of the 802.11 wireless communication protocol, and how they benefit accurate and low power jamming.

2.1 CSMA protocols and Wi-Fi

As defined by Kurose and Ross in [12], carrier-sense multiple access (CSMA) protocols employ random algorithms in order to access the data-link layer when a number of devices share the same medium to communicate. Common examples of CSMA protocols are the wired IEEE 802.3 Ethernet protocol [13] and the wireless IEEE 802.11 Wi-Fi protocol [14].

According to the CSMA protocol specification, before a station transmits, it first goes through a **carrier sense** phase, where it senses the status of the medium. When the medium is sensed as busy, the station will wait for a certain amount of time (Distributed Coordination Function Interframe Space, or DIFS, in the Wi-Fi protocol) until the channel becomes free again before transmitting. This precaution by itself does not guarantee collision-free access to the medium due to propagation delays and the Hidden Terminal Problem [12]. In order to ensure a transmitted Wi-Fi packet was indeed received without errors, the receiving station sends an acknowledgment frame (ACK) every time a packet is received without errors. If the transmitting station does not receive an ACK for a transmitted frame, it will attempt to re-transmit it several times until it gets acknowledged by the receiving station. If, after several attempts, the sending physical layers does not receive an ACK, it will discard the packet and alert the higher layer protocols [1].

The Wi-Fi protocol provides two different methods of detecting whether a channel is busy or clear. As specified in Subsection 17.3.10.6 of the 2016 version of the Wi-Fi standard[14], the Wi-Fi station performs both a carrier sense-based clear channel assessment (CS/CCA), in which it looks for Wi-Fi traffic on the channel, and a generic energy-detection based assessment (CCA-ED), in which it looks for any kind of energy on the Wi-Fi band.

2.2 Jamming and the sensitivity of CSMA protocols

Jamming is a Denial-of-Service (DoS) attack that makes advantage of a shared medium in electronic communication. Jamming is done by transmitting signals that interfere with the ability to communicate on the medium.

Stations communicating by the 802.11 protocol will wait until the medium is free for the DIFS time period in order to transmit their pending frames. Hence, if a transmitter continuously transmits on a channel in one of the 802.11 frequency bands, all the devices communicating on this channel will not be able to transmit. This kind of jamming will cause a denial of service to any wireless networks using that channel, usually causing these devices to indicate an error condition and ultimately abandon the channel altogether.

The kind of jamming described will even work for protocols that are not built on CSMA simply because in wireless transmission, the communication medium is shared between all. In wireless communication, the Signal-to-Noise Ratio (SNR) needs to be high enough for stations to be able to differentiate the transmitted data from background noise.

802.11 is also sensitive to low-power jamming, which has less of an effect on the SNR. As mentioned before, a station following the 802.11 protocol will not transmit in the presence of an existing transmission on the medium. By that observation, there is no need to overwhelm the existing traffic on the channel, but only to make the jamming signal powerful enough to trigger the CCA-ED mechanism. If the jammer is capable of creating properly-formed Wi-Fi packets, he can trigger the CS/CCA mechanism as well and improve his chances of jamming the channel.

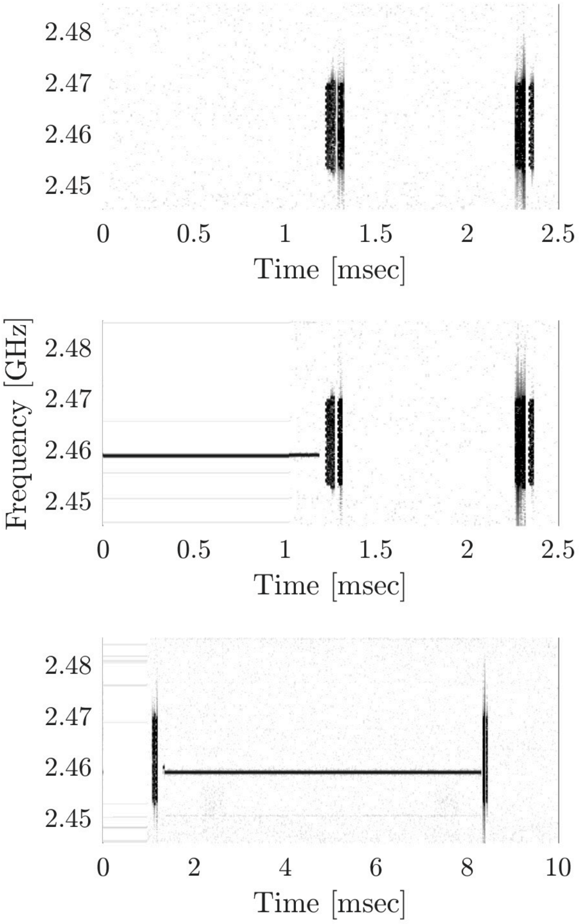


Figure 2: Top: a DNS query and response without jamming. Middle: DNS query delayed by micro-jamming. Bottom: DNS response delayed by micro-jamming.

2.3 Principles of micro-jamming

Micro-jamming is a technique shown and used in this paper that creates delays in the frames transmitted over a wireless networks. In contrary to traditional jamming techniques, which generate various degrees of DoS, micro-jamming does not aim to perform DoS. Instead, in a micro-jamming system, the jamming is switched on and off in high frequency. Because communications are not entirely blocked, transmitted frames may be delayed or re-transmitted, but they are rarely discarded. As a result, the impact of this method on the usability and

[1] An optional RTS/CTS access mode also exists for Wi-Fi, but it is not commonly deployed and will not be discussed here.

throughput of the overall channel is minimized.

An example of the operation of micro-jamming can be found in Fig. 2, which contains actual recordings of network traffic between a laptop and a wireless router, as captured using a Tektronix RSA604 real-time signal analyzer. The top part of Fig. 2 shows a simple DNS transaction carried out over Wi-Fi. The transaction begins when a higher layer of the protocol stack creates a packet (in this case, a DNS query), and asks the Wi-Fi physical layer to transmit it. The Wi-Fi controller on the laptop performs a clear channel assessment, which immediately succeeds in this case, and transmits the packet over the air, as indicated by the first energy band on the left. Very shortly after this event, the Wi-Fi physical layer on the router successfully receives the packet, and immediately sends an ACK packet back to the laptop. In the Figure this can be seen as a more powerful energy band which immediately follows the laptop's packet. The router next performs whatever operations it requires to satisfy the DNS query at the upper protocol level, either by handling it locally or by sending it to another machine. After the DNS response is ready, it is now the router's time to perform clear channel assessment, after which it sends a DNS response back to the laptop. This DNS response is indicated in the figure as a higher-energy band. Finally, the laptop receives this DNS response and immediately acknowledges it.

The middle part of Fig. 2 shows an actual micro-jamming scenario, in which the higher-level protocol layers request the Wi-Fi stack to transmit a DNS query while the shared medium is under the effect of micro-jamming, seen in the figure as a thin solid line. As soon as the micro-jamming stops, the laptop's physical layer waits for an additional short time (DIFS) and then immediately transmits the DNS query it has stored, and then converging with the previous case. It is important to note that in this case the application layer on the laptop will detect a larger round-trip delay before receiving the DNS response, since the DNS query was not immediately sent to the shared medium.

The bottom part of Fig. 2 shows a third scenario, in which the laptop was able to transmit the DNS query to the router as soon as it arrived from the higher protocol layers, but the router discovered that micro-jamming was being performed as it was preparing to send its answer to the laptop. As seen in the figure, the router delays sending its DNS response for as long as jamming is engaged. This can be seen in the figure as the extended solid line separating the DNS request (and associated ACK) and the DNS response (again with associated ACK).

As seen in the figure, micro-jamming is controlled by two parameters, the first is the frequency f_J in which the jamming is switched on and off, and the other is the jamming duty cycle D_J, that is, the proportion of the time for each cycle in which jamming is performed on the medium. We determine good values for both parameters in the following section.

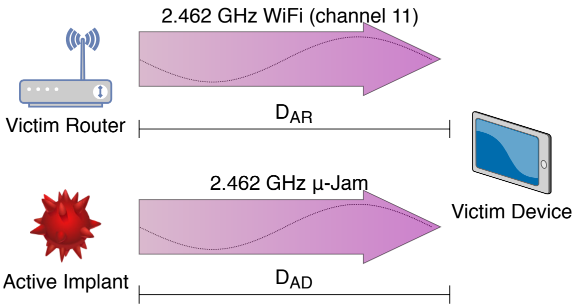


Figure 3: Active micro-jamming setup.

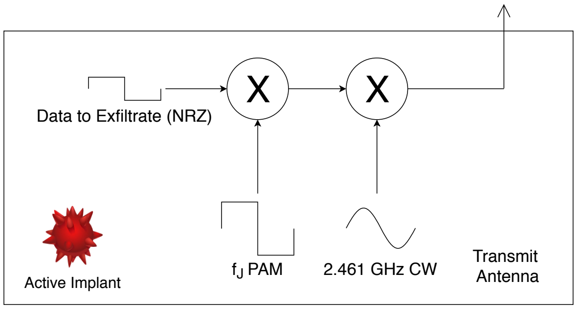


Figure 4: Design of the active Wi-Fi micro-jamming implant.

3 Active Jamming

As shown in Fig. 1, our model is constructed of a malicious implant acting as the data sender, an attacker acting as the receiver in a distant Internet connected location, and a victim device that is browsing a website with some attacker-controlled content using a Wi-Fi access point.

The implant's high-level diagram is shown in Fig. 4. The data to be exfiltrated is first multiplied with a square wave at frequency f_J, to create short breaks in the jamming signal that allow some traffic to go through. The resulting waveform is then used to modulate a sine wave at the central frequency of Wi-Fi channel 11, and finally amplified and transmitted over the air.

On the receiver side, the target device is induced to load attacker-controlled web content which includes a small amount of JavaScript code. This code causes the target device to issue Domain Name Service (DNS) requests to non-existent domains, and then measures the time it takes for an error response to arrive from the DNS server. This is done in practice by creating an HTML

Form Factor	Vendor	Model	OS	Browser
Phone	Samsung	Galaxy S7	Android 8.0	Chrome
Phone	LG	Nexus 5x	Android 8.1.0	Dolphin
Phone	Xiaomi	Redmi Note 4	Android 7.0	Mobile Browser
Phone	Apple	iPhone SE	iOS 11.3.1	Mobile Safari
Phone	Apple	iPhone 5	iOS 10.3.3	Chrome
Phone	Samsung	Galaxy S8	Android 8.0	Chrome
Phone	Samsung	Galaxy note 8	Android 8.0	Chrome
Tablet	Sony	Xperia	Android 5.1.1	Chrome
Laptop	Dell	Inspiron 5559	Windows 10	Chrome
Laptop	Apple	Macbook Pro	MacOS 10.13.4	Safari
Desktop PC	Lenovo	ThinkCentre 3212	Windows 10	Firefox
IoT Node	Raspberry Pi Foundation	Raspberry Pi 3	Raspbian Stretch	Chromium

Table 1: Hardware evaluated with the active implant.

Algorithm 1 Simplified JavaScript code for measuring network RTT based on DNS errors

```
function nextMeasurement() {
  var nextImage = "https://" +
    randomDomain(domainSuffix) + "/X.png";
  startTime = performance.now();
  document.getElementById('badImage').src =
    nextImage;
}

function onImageError() {
  var elapsedTime =
    performance.now() - startTime;
  elapsedTimes.push(elapsedTime);
  nextMeasurement();
}
```

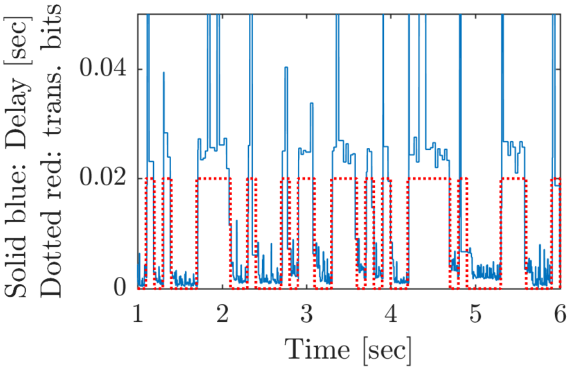


Figure 5: Waveform transmitted by the active implant (dotted red) alongside with the delays perceived by the JavaScript code running on the victim device (solid blue).

image element and setting its source to an image hosted on an invalid domain. We chose this method of probing the network since DNS requests require only one UDP packet per request and one packet per error response, in contrast to common web traffic which is carried over TCP and includes an additional round trip for connection setup. A simplified version of our code can be found in Listing 1.

When the malicious implant is wants to send a logical "1" to the victim device, it activates micro-jamming on the Wi-Fi channel, causing an increase in the round-trip time that is measurable in JavaScript.

3.1 Results

For the actual implementation of the attack we used an Atmel ATMEGA256RFR2 Xplained Pro board as the implant [15]. The ATMEGA256RFR2 microcontroller

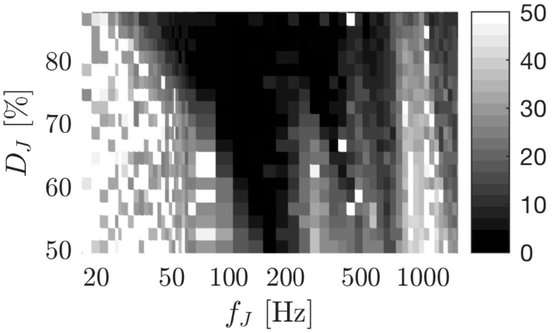


Figure 6: Bit error rate as a function of the micro-jamming frequency f_J and the duty cycle D_J at 30 bits per second.

is capable of generating raw unmodulated signals at the 2.4 GHz frequency band at a variety of frequencies and output levels. It has a unit price of less than $5, which compares well to other implants designs based on repurposed smartphones [16] or Wi-Fi dongles [17].

A small program compiled for the implant and allowed turning the jamming signal on and off by toggling one of the board's external I/O pins. We connected a Keysight 33622A 120 MHz Waveform Generator to this external pin and used it to choose different modulation patterns for the ATMEGA board. For the receiver we used a wide variety of different devices, as listed in Table 1, including phones, laptops, tablets and even an IoT node. All of the devices we evaluated were susceptible to the micro-jamming attack. A TP-Link WR940N 450Mbps Wireless N Router operated as the victim Wi-Fi access point, and a Raspberry Pi 3 running dnsmasq functioned as the local DNS server of the LAN created by the TP-Link Router.

JavaScript code was embedded in a locally hosted webpage. The script periodically sends DNS requests to non-existent URLs , effectively measuring the round-trip time (RTT) of the DNS message. When the channel is being interfered with, this RTT is comparatively higher than when no interference is present.

The experiment was run using various configurations for bit-rate, micro-jamming frequency (f_J) and duty cycles (D_J). Delays inflicted on the DNS queries could be easily translated into binary sequence when an effective configuration was applied, as shown in Fig. 5. Fig. 6 shows the affect of choosing various jamming frequencies and duty cycles on the bit error rate of the covert channel. When D_J is low there is higher likelihood that some of the DNS request will travel with no delay, making the exfiltration unreliable. The same is also correct for low f_J values. f_J values that are too high might not allow packets to be sent because the channel won't be free for a long enough duration to allow transmission.

Data rates of 5, 10, 20, 30 and 40 bits per second were tested against micro-jamming frequency $f_J = 16Hz - 1521Hz$ and duty cycles $D_J = 50\% - 88\%$. We found the lowest error rate was achieved with a duty cycle of 80% - 88%. the micro-jamming frequency had a great effect on bit error rates, as can be seen in Figure 7.

3.2 Range and transmission power

We measured the maximum distance at which the active attack can be deployed in three different scenarios. In the first scenario, the implant (sender) was located along the line between the device (receiver) and the router. In this scenario the attack worked at the maximum range at $D_{AR} = 32m$ and $D_{AD} = 15m$ (D_{AR} and D_{AD} as shown in Fig. 3). In the second scenario, we placed the device and the router $1m$ apart, then placed the implant some distance away from the two. In this scenario the attack was successfully with maximum distance of $D_{AR} = D_{AD} = 9m$. In the third scenario, we again placed the device and the router 1m apart, then attempted to carry out the attack from the other side of an internal wall. In this scenario the attack was successful with the maximum range of $D_{AR} = D_{AD} = 3m$. The power level of the implant was set to its maximum level of 3dBm in all cases.

We next measured the minimum power level at which the attack could still be successful. The lowest power level available from our active implant was -17 dBm, equivalent to 20 microwatts. Even at this reduced power level, data transfer between the implant and the target device was still possible, albeit at a reduced range of $D_{AR} = D_{AD} = 0.5m$.

3.3 Effect of duty cycle on throughput

In order to quantify the effect micro-jamming has on network behavior, an experiment was planned where the victim device downloads a large file over the network while micro-jamming is active in different duty cycles.

Fig. 8 shows the average throughput during active micro-jamming. While increasing jamming duty-cycle evidently leads to a drop in network performance, even a 95% duty-cycle, leaves the network in a highly functioning state.

4 Passive Jamming

This section describes the fundamentals of Wi-Fi backscatter and shows a micro-jamming device built using the methods of Kellogg et al. [6, 18]. The device is capable of performing data exfiltration similarly to what

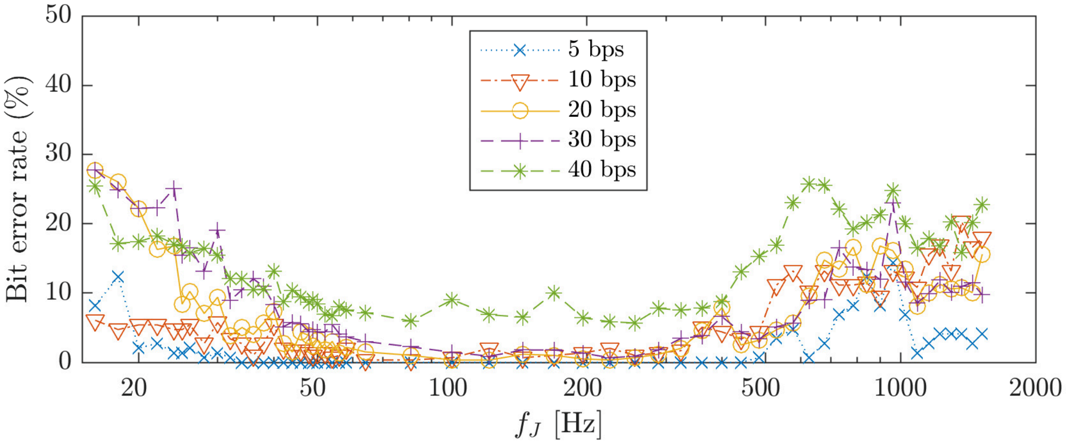


Figure 7: Best bit error rate as a function of frequency for active micro-jamming.

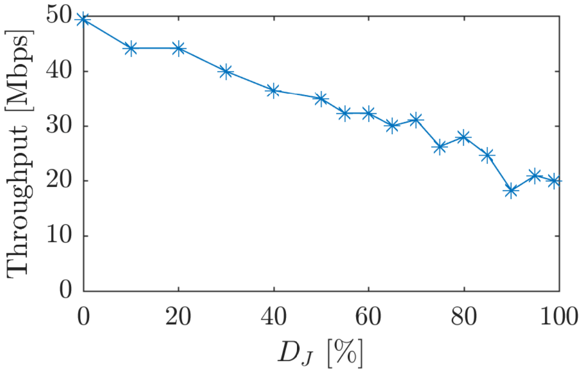


Figure 8: Effect of duty cycle D_J on throughput.

had been shown in previous sections without consuming any power for transmission.

4.1 Backscatter theory

Radio backscatter is a known physical phenomenon that occurs when a passive antenna switches its impedance on a certain frequency. The result is a periodical alternation of the antenna's radar cross section and hence, its reflectance of radio signals. Signals in the form of electromagnetic waves are reflected from the antenna periodically causing a shift in the frequency of the reflected wave [6].

The affect of modulating the antenna's impedance in the presence of electromagnetic radiation can be seen as the multiplication of two sinusoidal signals: the carrier wave frequency f_{CW}, and the antenna's frequency f_{MOD}. The known trigonometrical identity $2sin(f_{CW})sin(f_{MOD}) = cos(f_{CW} - f_{MOD}) - cos(f_{CW} + f_{MOD})$ shows that a frequency shift will form within the distance of $\pm f_{MOD}$ from the carrier wave frequency f_{CW}. This effect is used in Radio Frequency Identification (RFID) devices [19] and other low power applications. Usage of backscatter for Wi-Fi data transmission was also studied and found to be viable [6, 20].

Applied to our problem, we attempted to build an implant which does not directly transmit a signal at the jamming frequency, but instead takes an existing radio signal from the environment and modulates it into the channel to be jammed. Following the work of [6], we investigated this setup in two different settings. The first setting is **semi-passive backscatter**, which requires a designated carrier wave to be transmitted for the backscatter to succeed. The carrier wave frequency is outside of the backscatter transmission channel boundaries, and therefore does not interfere with the channel. When backscatter modulation is enabled, however, this signal is shifted by the passive antenna, resulting in a signal within the channel boundaries which should cause a jamming effect. The second and more challenging setting is that of **fully-passive backscatter**. Fully-passive backscatter does not assume a designated carrier wave source, but instead relies on existing electromagnetic communication for the generation of a new signal. In theory, as long as there are some existing Wi-Fi transmissions in adjacent channels, a passive antenna can be used to frequency shift these signals into the targeted channel and cause a jamming effect.

4.2 Equipment

The setup for performing passive Wi-Fi micro-jamming was built of the following components:

The passive backscatter implant is described in Figure 9. It was designed around a HMC190BMS8 GaAs MMIC SPDT integrated circuit; an electronic circuit was

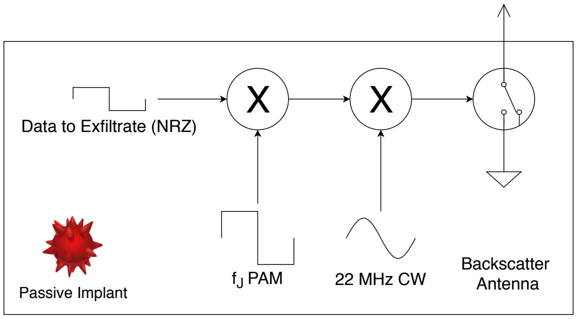


Figure 9: Design of the passive Wi-Fi micro-jamming implant.

prepared using the design of the evaluation circuit board seen in the datasheet [21]. One output of the circuit board was connected to the electrical ground while the other left floating, similar to the design of Kellogg et al. [6, 18]. The implant was switched between an open and closed circuit mode at a switching frequency of 22 MHz using a Keysight 33622A waveform generator. To generate a data signal, the modulation input of the waveform generator was controlled by a PicoScope PS6404D arbitrary waveform generator. A schematic of the implant can be seen in Fig. 9. The antenna attached to the implant was an off-the-shelf directional antenna with a 10cm ∅ metal reflective dish.

We evaluated two different scenarios: in the **semi-passive setting** the Atmel ATMEGA256RFR2 Xplained Pro development board was fitted with a directional antenna built of an aluminum can (this design is popularly known as a 'Cantenna' [22]). The transmitter was configured with transmission strength of 3.2dBm and set to transmit a carrier wave at Wi-Fi channel 6 (2.437 GHz). In the **fully passive setting** an additional TP-Link WR940N router was set up to use Wi-Fi channel 6 (2.437 GHz). A laptop was connected to this router via Wi-Fi and was configured to generate constant network traffic by repeatedly downloading a large file.

The passive jamming setup, including the passive transmitting implant and the directional antenna, is shown in Fig. 10.

4.3 Parameters

The victim device was connected to the router on 802.11 channel 11 that has a 22Mhz span and is centered on the 2462MHz frequency. For the semi-passive experiment, a carrier wave of 2437MHz was directionally transmitted to the passive implant. The carrier wave was transmitted with the intensity of 3.2dBm. For the fully-passive experiment, a second router was connected to a laptop over Wi-Fi channel 6 (centered on the 2437MHz frequency)

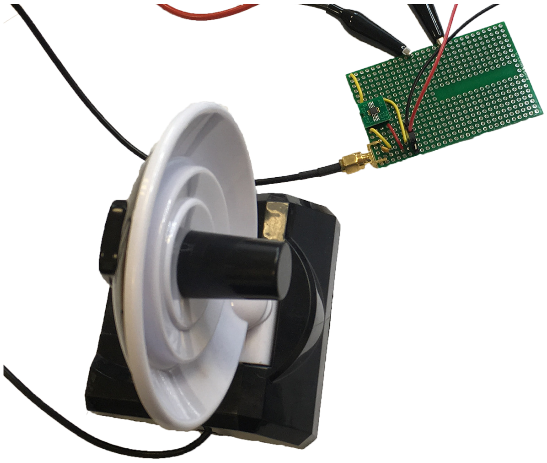
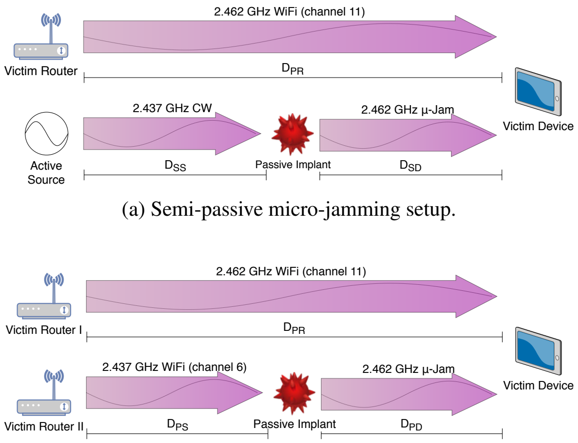


Figure 10: Passive jamming setup



(a) Semi-passive micro-jamming setup.

(b) Fully-passive micro-jamming setup.

Figure 11: Passive micro-jamming setup.

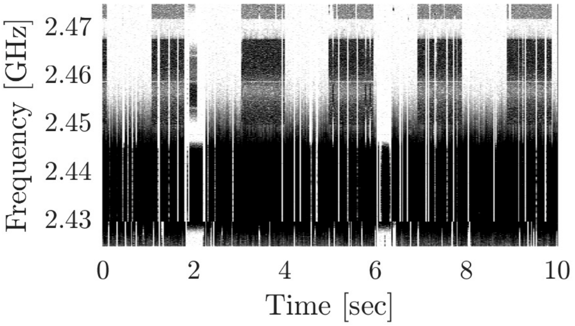


Figure 12: Spectrogram of passive micro-jamming.

that was downloading a large file from the network. The passive implant was switched at a frequency of 22 MHz and aimed at the victim device.

Entities participating in the jamming were arranged in a manner that imitates real-life scenarios where the locations of the passive implant and carrier wave transmitters are independent of the router position. A Tektronix RSA306 real-time signal analyzer was placed next to the device antenna for capturing and analyzing the signals.

Diagrams showing the test parameters can be seen in Fig. 11.

4.4 Viability

4.4.1 Semi-passive experiment

Using the experimental setup described above, the device was observed experiencing repeating delays in DNS queries sent by the JavaScript code. The signal analyzer captured a modulated 2456Mhz transmission in addition to the 2437 MHz carrier wave. The power of the 2456Mhz transmission generated by the passive antenna was measured to be approximately 20dBm less than the carrier wave, depending on the position of the antennas.

The test was performed at 18.3 bits per second and measured bit error rate of 0.0056 while the CW transmitter and passive antenna were in close range.

4.4.2 Fully passive experiment

During the fully passive experiment, transmissions from Wi-Fi channel were seen to be backscattered into channel 11 and the device connected to channel 11 was experiencing network delays when the jamming was active. Fig. 12 shows the spectrum output of this experiment, as captured by the real-time signal analyzer. As shown in the figure, when the passive micro-jammer was active, the power output of the Wi-Fi endpoints on channel 6 was backscattered into channel 11. It is worth mentioning that due to the nature of backscattering, during the time while transmissions from channel 11 were backscattered into channel 6, transmissions from channel 6 were also backscattered into channel 11.

5 Discussion

The attacks shown in this paper were proven to work with several different victim devices and we deduce that all Wi-Fi capable devices are susceptible to these kind of attacks. Moreover, the power requirements are very low, with the active transmitter consuming less than 20 microwatts for transmission, power levels which are low enough to be realized in miniature battery-operated implants. The semi-passive or fully-passive implants shown in the paper can have even lower transmission power consumption.

The implants described require no physical contact and can also operate when walls are separating them from the devices. We make the observation that while many jamming techniques require a signal more powerful than the jammed signal, our technique does not. In fact, minimal reception of the jamming signal is enough to cause interruption in the Wi-Fi communication.

The interplay between different layers in the networking stack is interesting: the jamming uses raw unmodulated radio signals, therefore it can be classified as a layer 0 attack. However, its transmit power level is far too low to overcome 802.11 transmissions at the physical layer (layer 1). Instead, it works by exploiting the oversensitivity of the 802.11 MAC protocol (layer 2) to in-channel interference. While the sender uses layer 0 signals to perform the attack, the receiver actually uses an untrusted JavaScript application to issue multiple DNS requests (at layer 7).

In contrast to sensor-based attacks, which can be addressed by disallowing access to sensors, webpages and the scripts within them must be allowed to use the network for even the most basic of functions. This makes defending against this exfiltration method very difficult. One approach for mitigation could be to delay or throttle the amount of image onerror and similar messages provided to a webpage. However, this would only reduce the bit-rate of the attack and not overcome it completely.

A positive use for micro-jamming can be in ZigBee to Wi-Fi communications, since typically an IoT device, such as a smart lamp, has limited communications abilities. Specifically, while the device may not have a Wi-Fi compatible radio, in many cases its ZigBee radio hardare is still capable of generating unmodulated carrier wave signals in the 2.4 GHz frequency band [23]. As long as the IoT device uses a ZigBee channel which is shared with a Wi-Fi channel, it can use micro-jamming to communicate with a webpage running on the PC. This allows for a new communication channel to be used in diagnostics and status reporting.

5.1 Conclusion

In this paper, we introduced micro-jamming as a way to create delays in packet sending at the 802.11 protocol without causing a loss of data. Using micro-jamming, a covert channel was created by abusing a device connected to Wi-Fi (victim) and using a low powered transmitter (implant). Following these demonstrations, we also implemented micro-jamming using semi-passive and passive transmitters that harnesses its required transmission energy from existing background electromagnetic transmissions. The covert channel created by applying these techniques requires no sensors or extra permissions, achieves a practical bit rate and error rate under reasonable operating conditions, and can be made very difficult to detect.

Acknowledgments

This research was supported by Israel Science Foundation grants 702/16 and 703/16. We thank Eyal Ronen for providing the inspiration and equipment for starting off this research.

References

[1] Gaurav Shah and Matt Blaze. Covert channels through external interference. In Dan Boneh and Alexander Sotirov, editors, *3rd USENIX Workshop on Offensive Technologies, WOOT 09, Montreal, Canada, August 9, 2009*, pages 1–7. USENIX Association, 2009.

[2] Genesis 3:8.

[3] Mark T Hove. *History of the Bureau of Diplomatic Security of the United States Department of State*. US Department of State, Bureau of Diplomatic Security, 2011.

[4] Benyamin Farshteindiker, Nir Hasidim, Asaf Grosz, and Yossi Oren. How to phone home with someone else's phone: Information exfiltration using intentional sound noise on gyroscopic sensors. In Natalie Silvanovich and Patrick Traynor, editors, *10th USENIX Workshop on Offensive Technologies, WOOT 16, Austin, TX, August 8-9, 2016.* USENIX Association, 2016.

[5] David E Sanger and Thom Shanker. NSA devises radio pathway into computers. *The New York Times*, 1(15), Jan 2014.

[6] Bryce Kellogg, Vamsi Talla, Shyamnath Gollakota, and Joshua R. Smith. Passive wi-fi: Bringing low power to wi-fi transmissions. In Katerina J. Argyraki and Rebecca Isaacs, editors, *13th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2016, Santa Clara, CA, USA, March 16-18, 2016*, pages 151–164. USENIX Association, 2016.

[7] Lennart Haagsma. Deep dive into QUANTUM INSERT. Online at https://blog.fox-it.com/2015/04/20/deep-dive-into-quantum-insert/.

[8] Weiwei Jiang, Denzil Ferreira, Jani Ylioja, Jorge Gonçalves, and Vassilis Kostakos. Pulse: low bitrate wireless magnetic communication for smartphones. In A. J. Brush, Adrian Friday, Julie A. Kientz, James Scott, and Junehwa Song, editors, *The 2014 ACM Conference on Ubiquitous Computing, UbiComp '14, Seattle, WA, USA, September 13-17, 2014*, pages 261–265. ACM, 2014.

[9] Michael Hanspach and Michael Goetz. On covert acoustical mesh networks in air. *JCM*, 8(11):758–767, 2013.

[10] Luke Deshotels. Inaudible sound as a covert channel in mobile devices. In Sergey Bratus and Felix F. X. Lindner, editors, *8th USENIX Workshop on Offensive Technologies, WOOT '14, San Diego, CA, USA, August 19, 2014.* USENIX Association, 2014.

[11] Chromium Security Team. Deprecating powerful features on insecure origins. Online at https://www.chromium.org/Home/chromium-security/deprecating-powerful-features-on-insecure-origins.

[12] James F. Kurose and Keith W. Ross. *Computer Networking: A Top-Down Approach (6th Edition)*. Pearson, 2012.

[13] Robert Metcalfe and David Boggs. Ethernet: Distributed packet switching for local computer networks. *Commun. ACM*, 19(7):395–404, 1976.

[14] IEEE 802.11 Working Group. 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) specifications. *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, pages 1–3534, Dec 2016.

[15] Microchip Technology Inc. Atmega256rfr2 xplained pro evaluation kit. `http://www.microchip.com/DevelopmentTools/ProductDetails.aspx?PartNO=atmega256rfr2-xpro`.

[16] Matthias Schulz, Francesco Gringoli, Daniel Steinmetzer, Michael Koch, and Matthias Hollick. Massive reactive smartphone-based jamming using arbitrary waveforms and adaptive power control. In Guevara Noubir, Mauro Conti, and Sneha Kumar Kasera, editors, *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec 2017, Boston, MA, USA, July 18-20, 2017*, pages 111–121. ACM, 2017.

[17] Mathy Vanhoef and Frank Piessens. Advanced wifi attacks using commodity hardware. In Charles N. Payne Jr., Adam Hahn, Kevin R. B. Butler, and Micah Sherr, editors, *Proceedings of the 30th Annual Computer Security Applications Conference, ACSAC 2014, New Orleans, LA, USA, December 8-12, 2014*, pages 256–265. ACM, 2014.

[18] Bryce Kellogg, Aaron N. Parks, Shyamnath Gollakota, Joshua R. Smith, and David Wetherall. Wifi backscatter: internet connectivity for rf-powered devices. In Fabián E. Bustamante, Y. Charlie Hu, Arvind Krishnamurthy, and Sylvia Ratnasamy, editors, *ACM SIGCOMM 2014 Conference, SIGCOMM'14, Chicago, IL, USA, August 17-22, 2014*, pages 607–618. ACM, 2014.

[19] Daniel M. Dobkin. *The RF in RFID, Second Edition: UHF RFID in Practice*. Newnes, 2012.

[20] Dinesh Bharadia, Kiran Raj Joshi, Manikanta Kotaru, and Sachin Katti. Backfi: High throughput wifi backscatter. In Steve Uhlig, Olaf Maennel, Brad Karp, and Jitendra Padhye, editors, *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, SIGCOMM 2015, London, United Kingdom, August 17-21, 2015*, pages 283–296. ACM, 2015.

[21] Analog Devices. *HMC190BMS8 / 190BMS8E Datasheet*. `http://www.analog.com/media/en/technical-documentation/data-sheets/hmc190b.pdf`.

[22] Olga Saukh, Robert Sauter, Jonas Meyer, and Pedro José Marrón. Motefinder: A deployment tool for sensor networks. In *Proceedings of the workshop on Real-world wireless sensor networks*, pages 41–45. ACM, 2008.

[23] Eyal Ronen, Adi Shamir, Achi-Or Weingarten, and Colin O'Flynn. Iot goes nuclear: Creating a zigbee chain reaction. *IEEE Security & Privacy*, 16(1):54–62, 2018.