FOCUS: Frequency Based Detection of Covert Ultrasonic Signals

Peer-reviewed author version

HELLEMANS, Wouter; RABBANI, MD Masoom; VLIEGEN, Jo & MENTENS, Nele (2022) FOCUS: Frequency Based Detection of Covert Ultrasonic Signals. In: Meng, Weizhi; Fischer-Hübner, Simone; D. Jensen, Christian (Ed.). ICT SYSTEMS SECURITY AND PRIVACY PROTECTION (SEC 2022), SPRINGER INTERNATIONAL PUBLISHING AG, p. 70 -86.

DOI: 10.1007/978-3-031-06975-8_5
Handle: http://hdl.handle.net/1942/39313

FOCUS: Frequency based detection of Covert Ultrasonic Signals

Wouter Hellemans[1], Md Masoom Rabbani[2], Jo Vliegen[2], and Nele Mentens[2,3]

[1] KU Leuven & University of Hasselt, Belgium
wouter.hellemans@student.uhasselt.be
[2] ES&S, imec-COSIC, ESAT, KU Leuven, Belgium
{mdmasoom.rabbani,jo.vliegen,nele.mentens}@kuleuven.be
[3] LIACS, Leiden University, The Netherlands
n.mentens@liacs.leidenuniv.nl

Abstract. Today's evolving and inventive attacks allow an adversary to embed tracking identifiers or malicious triggers in ultrasonic sound and covertly transmit them between devices without the users' knowledge. An adversary can exploit an electronic device by manipulating the microphone, gyroscope or speaker using ultrasonic sound. Almost all types of electronic devices are vulnerable to this type of attack. Indeed, some preventive measures are in place to counter ultrasonic invasion. However, they are primitive and often are not capable of detecting the attacks.
To this end, we propose FOCUS: Frequency based detection of Covert Ultrasonic Signals. In particular, FOCUS displays a low-end, low-cost ultrasonic detection mechanism that can be employed anywhere. We validate FOCUS through two proof-of-concept (PoC) implementations utilizing Raspberry Pi and Arduino based hardware modules, respectively. The results demonstrate that FOCUS can detect ultrasonic sound and alert users of possible ultrasonic invasion.

Keywords: Embedded system, Ultrasonic sound invasion, Network Security & Privacy

1 Introduction

Collecting information about users is becoming an ever more important part of the business strategy of various companies. The increasingly stringent regulations (e.g., the European GDPR[1]), have caused companies to use new and controversial strategies to collect this information about the user. One of these emerging technologies is ultrasonic invasion (UI). This technology uses ultrasonic beacons (UB) that are imperceptible to humans[2], but are sensed by mobile devices.

The main application of UB lies in cross-device tracking (xDT). The purpose of xDT is to establish a profile of the user across different devices. Typically,

[1] https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu_nl
[2] The upper hearing threshold in practice is around 17 kHz. Thus the near-ultrasonic range is 17-20 kHz

xDT is done by exchanging unique identifiers between different devices (e.g., an advertising identifier). This poses a serious privacy threat since, by targeting different devices, a much more detailed profile can be created than with traditional tracking technologies (e.g., Cookies). For example, with the use of xDT by means of UB, it becomes possible to link a work phone with a personal phone when they are on the same desk. In addition, since the entire process of receiving the signals emitted from the beacons by electronic devices usually happens in the background, the users are often unaware of the profile that is being created.

Since API level 26, Android has placed several restrictions[3] on background services, which make the unnoticed processing of UB signals more challenging. Because these background services consume device resources without the user being aware, they could result in a deteriorated user experience. Therefore, they are killed by the Android operating system after 1 minute. For long running services, the concept of a foreground service was introduced. All this makes receiving UB signals more difficult, yet not impossible. After all, it is possible to do the receiving of the signals emitted from the beacons with a background service within the 1 minute timeframe. Further, the receiving of UB signals can also be built into an unsuspicious foreground service of an application (e.g., a music application).

Another measure that makes the sensing of UB signals less stealthy on an Android mobile device, is the run-time permission required to record audio. Since API level 23, the RECORD_AUDIO permission of the MediaRecorder API is considered a 'dangerous' permission[4]. Whereas normal permissions are validated at application installation, the Android operating system forces dangerous permissions to be validated at-run time. However, when UB sensing is for example built into an application that supports voice messaging, the users are most likely unaware that permitting the audio record poses a threat to their privacy. Similar concepts apply to iOS, but are not discussed further in this paper.

What makes this research even more urgent, is the fact that ultrasonic technology is already being actively deployed. Lisnr [4] is a company, with over 250'000 customers, that offers services using UB (e.g., for contactless transactions using UB between mobile wallets and vendors). Another example is the company Shopkick, where users can collect points for vouchers by walking into participating stores that have ultrasonic transmitters installed. Since the previous two companies only do beacon sensing while their application or website is open on the user's mobile device, the users are aware that their microphone is being used. However, there are also companies that run the entire process in the background. Silverpush is a company that has developed services to embed UB sensing as third-party content in, for example, a website or application. In this way, a user only needs to have an application with a Silverpush component installed on his device and the personal information, coming from UB embedded in TV streams, websites or even billboards is completely collected and processed in the background. In 2017, researchers discovered 234 apps with Shopkick com-

[3] https://developer.android.com/about/versions/oreo/background
[4] https://developer.android.com/guide/topics/media/mediarecorder

ponents listening to UB in the background [1]. In this paper, we introduce and compare three different ways to detect sounds in the near-ultrasonic range and alert the user of the presence of these signals. We call our solution 'Frequency-based detection of Covert Ultrasonic Channels (FOCUS)'.

Our solution provides the following contributions:

1. To the best of our knowledge, FOCUS is the first work to consider the external detection of signals in the near-ultrasonic range. This external detection intercepts the signal, allowing users to detect the attack, even before it reaches their device. This opens up the path to external mitigation mechanisms that prevent the ultrasonic signals from reaching or influencing the victim device.
2. FOCUS runs completely in the background of the user's device, allowing the detection of ultrasonic sound without any interaction.
3. FOCUS operates completely wireless, allowing the detector to be installed in hard-to-reach places.
4. We present two PoC implementations, based on Arduino and Raspberry Pi hardware modules, and we show that the results are promising. Because of the limited resources in the Arduino module, standard frequency analysis methods cannot be deployed. Therefore, we propose and implement an lightweight solution for detecting ultrasound signals on low-cost platforms.

The remainder of this article is organized in the following manner. We discuss the state-of-the-art on acoustic air-gapped covert channels in Section 2 and provide a background overview in Section 3. In Section 4, we give the problem setting and in Section 5, we discuss the system model and adversarial assumptions. Sections 6 and 7 discuss the protocol and the results obtained. In Sections 8 and 9, we discuss the advantages and disadvantages of our solution and give a security analysis. Finally, the paper is concluded in Section 10, where we also discuss the possibility of future research.

2 Related Work

A covert channel is a means of communication between two entities that was not anticipated by the designer of the entities. The concept of a covert channel was first formulated by Lampson in 1973 [11]. Since this type of communication is not intended to happen, covert channels can pose a serious privacy threat by bypassing existing communication protocols (e.g., for the purpose of data exfiltration). A special type of covert channels are air-gapped covert channels. What makes this type of covert channel more dangerous than other types, is the fact that it can occur between entities that are physically and logically disconnected from each other. The classification proposed by [8] divides air-gapped covert channels into five main categories: electromagnetic, magnetic, optical, thermal, and acoustic. However, in FOCUS, the main emphasis lies on the acoustic channel. Thus we discuss acoustic in the below section.

2.1 Acoustic

The covert channel that this paper focuses on, is the acoustic covert channel. This type of channel uses sound waves to exchange information between two computers. Depending on the type of sound used, two main categories can be distinguished: audible and ultrasonic acoustic covert channels.

Audible. This acoustic method uses a sound with a frequency that can be perceived by the human hearing (smaller than about 17 kHz). The concept was first explored by Madhavapeddy et al. [12], who conducted a comparative study between different modulation schemes to establish a speaker-to-microphone communication between two computers. Since this approach requires the transmitter to be equipped with a speaker, speakerless audible acoustic covert channels were later introduced by Guri et al.: Fansmitter uses noise from the fans of a PC for the transmitter [7]. DiskFiltration, as an alternative, uses noise coming from the actuator arm of a hard disk drive [9]. The main drawback that all these audible methods share is the fact that the noise can be perceived by the victim.

(Near)-Ultrasonic. This type of covert-channel, which uses signals in the 17-20 kHz range, is the main subject of this paper. Near-ultrasonic signals have the advantage that they are imperceptible to human hearing, yet can be generated and sensed by commodity hardware. This is because commodity microphones typically have a sampling frequency of 44.1 kHz. A first speaker-to-microphone implementation of this type of covert channel between computers was realized by Madhavapeddy et al. in 2005 [12] and could achieve speeds of 8 bps over a distance of 3.4 m. Later work by Carrara et al. achieved speeds of 230 bps, over a distance of 11 m, with a speaker-to-microphone implementation [5]. In 2015, Deshotels et al. accomplished a speaker-to-microphone ultrasonic covert channel between smartphones that could achieve a bit rate of 345 bps over a distance of 30 m [6]. All of these implementations share the disadvantage of requiring a microphone for detection. However, in secure environments, microphones are often not available due to security reasons. Therefore, Guri et al. proposed a microphoneless approach, in which a passive speaker is reversed by means of jack retasking and can therefore serve as a receiver [8]. Another microphoneless approach was presented in 2018 by Matyunin et al. Their research demonstrated that it is possible to use a MEMS gyroscope as a receiver for ultrasonic signals near the resonant frequency of the gyroscope (19-29 kHz) [13].

2.2 Differences with previous research

Whereas previous research has focused on developing a complete communication protocol (i.e. transmitter and receiver), this research focuses on the external detection of ultrasonic covert channels using a microphone. The main difference is that this research tries to intercept the ultrasonic signal with an external device, rather than having the detection happen on the victim's device, like in [13,8,12,6,5]). In this way, it is possible to intercept the sound before the victim has a chance to process it, opening up the path to external prevention mechanisms. In addition, developing both a transmitter and receiver has the

convenience that the receiver can already have information about the transmitted signal in advance (e.g., the frequency in [6]), which makes the sensing much easier but not applicable to any other general ultrasonic signal. Another point in which this study differs from previous ones is the fact that earlier research used spectral analysis for detection, whereas in this paper, a novel approach is proposed that does not require a high-end analog-to-digital converter (ADC). For example, [13] uses a fast Fourier transform (FFT) on the acquired data from the gyroscope. Similarly, [8] uses spectral analysis for detection (Praat tool[5]). On top of that, [8] requires the use of an invasive kernel driver to remap the audio jack to an input, while this paper focuses on minimal invasive approaches. Our proof-of-concept implementation on a low-end Arduino platform shows that the detection can successfully be done without the use of traditional frequency analysis mechanisms.

3 Background

This section provides background information on various types of modulation schemes that can be used in a near-ultrasonic communication protocols. Then, an overview is given of the algorithms typically used to decode the received ultrasonic signal on the victim's device. Finally, this section highlights an important limitation to the use of spectral analysis for the detection of ultrasound.

3.1 Possible modulation schemes

For the transmission of ultrasonic signals, several modulation schemes have been proposed. On-off keying (OOK) uses the presence or absence of a sine wave signal to encode a 1 or a 0, respectively. Although this approach is the simplest, in most cases OOK has a lower data rate than the other modulation schemes. Another modulation scheme is phase-shift keying (PSK), in which data are encoded by varying the phase of a signal of constant frequency. This scheme, however, has the disadvantage that discontinuities can occur in the ultrasonic signal, which can still produce an audible click [2] from the speakers. A final modulation scheme, that is used in practice by SilverPush and Lisnr [2], is frequency-shift keying (FSK). FSK encodes bits by changing the frequency of a carrier signal. The number of frequencies that can be used, and associated the data rate, are determined by the allowed frequency range and by noise. Like PSK, discontinuities can occur in the signal with this modulation scheme, resulting in audible clicks from the speakers. One solution to these clicks, which was proposed by Deshotels, is to gradually decrease the amplitude of the signal at the beginning and end of each transmitted character [6]. Since the modulation scheme employed depends on the attacker, this study considers any signal in the near-ultrasonic range as a potential covert channel. In addition, it is not known in advance which frequencies are used by the attacker, so the full range of 17-20

[5] https://www.fon.hum.uva.nl/praat/

kHz is to be considered. However, if the frequencies are known in advance, more efficient detection is possible.

3.2 Algorithms used for decoding ultrasound

Previous research indicated that Fast Fourier Transform (FFT) and Goertzel are the main algorithms for the detection of UB [2]. Both FFT and Goertzel are algorithms for efficiently computing the Discrete Fourier Transform (DFT) of a digital signal, whose output can be regarded as samples of the Discrete Time Fourier Transform (DTFT) of a digital signal. That is, the DTFT, which is a subcategory of the Z-transform, converts an input sequence in the time domain into an output sequence in the frequency domain. In general, the DTFT of an infinitely long sequence is given by:

$$H(\omega) = \sum_{n=-\infty}^{\infty} h(n)\, e^{-j\omega n}$$

Where H(ω) in general, is a complex number function of the angular frequency ω. For the detection of ultrasonic signals, only the magnitude of the complex number is of interest and the phase can be disregarded. As a result, the magnitude can be computed directly without separately calculating the real and imaginary parts. In this way, the computational efficiency of the algorithm is improved.

In practice, SilverPush would use only Goertzel and Shopkick would use only FFT. In addition, Lisnr would use both Goertzel and FFT. Whereas FFT has a lower time complexity to compute a range of frequencies, Goertzel has the advantage of detecting a single frequency with little computational effort. The latter is mainly important in mobile applications, where battery consumption plays an important role. In addition, a recent comparison of Goertzel and FFT for dual Tone Multi-Frequency (DTMF) detection showed that for smaller bin sizes, Goertzel gives more accurate results [10]. In this study, FFT failed to detect DTMF tones correctly for a bin size smaller than 128, while Goertzel's algorithm was successful. In this regard, Goertzel solves the reliability problem of external detection, which was mentioned by [8] as the main limitation of this approach.

3.3 Limitation of spectral analysis

The main disadvantage of the FFT and Goertzel algorithms for the detection of UB, is that they require a high-end analog-to-digital converter (ADC) according to the Nyquist theorem. To address this problem, this paper also proposes a simpler approach that does not require this ADC and uses of a digital counter.

4 Problem Setting

Wherever we go and whatever we might be doing, as long as we have an electronic device with a microphone, speakers or gyroscope, that device might be listening. All types of electronic smart devices (e.g., laptops, tablets, headphones, smartwatches, IoT home appliances) are working with the most clandestine and unavoidable methods for tracking our locations and behaviour.

Ad-tracking audio signals are used by mobile apps, which our phone can detect but we cannot. For example, we watch television as regular viewers and we have our phone(s) nearby or with us in the TV room. Presently, TVs are working as beacons that emit ultrasound and our phone(s) are working as receivers. Beacons emit high frequency sounds and receivers listen to them.

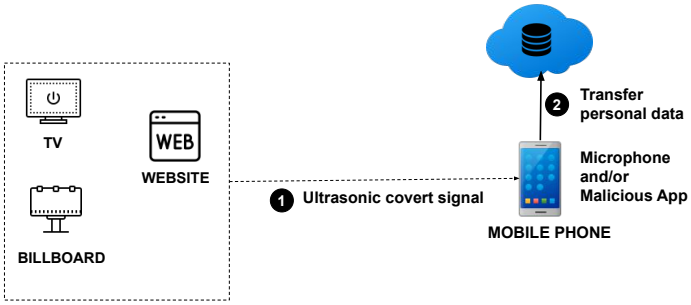

Fig. 1. Ultrasonic invasion through TV, websites, advertising boards

As shown in Figure 4, TVs, ultrasonic beacon embedded websites or ultrasound emitting ad boards will emit ultrasound during commercial breaks that we will not notice, but our phones and IoT appliances with microphones will. Our phones can create an identifier with details about us watching a specific show at a specific time after receiving signals from the TV. Our phone can save these details and share with various applications, which will then give it to third-party users. The core idea is to link several devices we own in order to locate us and gather information about us. Beacons may be incorporated into tablets, phones, websites, and even billboards. This whole process takes place in the background and does not require user permission.

5 System Details

In this section we describe our system model and adversarial assumptions.

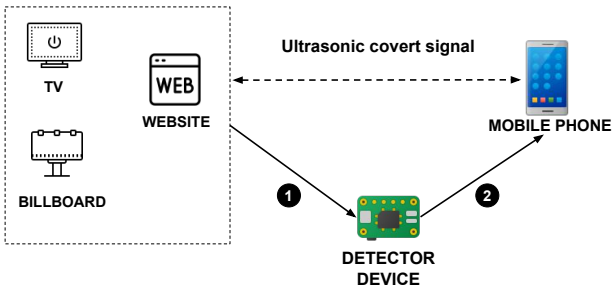

Fig. 2. System Model

System Model We consider a typical setting where a user has a mobile phone, laptop or desktop and TV or has access to an advertising billboard. In designing the ultrasonic detection scheme of such a system, we consider the presence of three main entities as shown in Figure 2.

- Ultrasound emitter: As shown in Figure 2, TVs, websites, billboards etc. are ultrasound emitters. Using the embedded ultrasound beacon, these devices emit ultrasound to gather information about users.
- Detector device: This device operates externally from the attacker and the victim, and detects the ultrasound emitted by the attacker. Furthermore, it alerts the victim of the malicious activity happening. In our PoC implementation we use Arduino and Raspberry Pi platforms as detector devices.
- Mobile device: The mobile device shown in Figure 2 serves a dual function. On the one hand, this device is the victim, and will transmit various personal data to a database upon detection of ultrasonic signals emitted by the attacker. On the other hand, the mobile device also constitutes a part of the proposed solution, in the sense that it communicates with the detector device to alert the user.

Adversarial assumptions We presume for our adversarial ability analysis that a victim (user) frequently uses a mobile phone, watches TV, and surfs the Internet. In terms of attacker capabilities, we look at previous research [15,3] as well as commercial implementations (Shopkick, Lisnr, Signal360, and Silverpush) that use microphones as receivers that can be exploited to track a victim's location or gain private information from the victim's mobile. Thus for attackers capabilities we assume three main attack scenarios:

- Web tracking: Ultrasonic beacons embedded in websites can emit ultrasound which can be received by the microphone of a victim's mobile.

- Commercial advertisement tracking: The adversarial media (e.g., TV or Billboard) provider uses the ultrasonic beacons with encoded tracking IDs that are embedded in broadcast content. The adversary can monitor what and when users watch by capturing these IDs with an application installed on the user's mobile device.
- Data theft: Adversary app components can be embedded as third party content in ignorant applications installed on different devices of the victim. We assume that there is at least one device with an infected app that can transmit, and at least one device with an infected app that can receive. In this way, sensitive personal data can be exchanged between different devices via ultrasonic communication in the background, and thus an aggregated profile of the victim can be established.

6 Our Protocol: Ultrasonic Invasion Detection

FOCUS is comprised of three different detection mechanisms, respectively utilizing low-end, mid-end, and high-end hardware. First, a low-end approach is presented that uses a digital counter as a software-based high-frequency detector implemented on an Arduino. Next, the mid-end approach is discussed that uses the Goertzel algorithm on a Raspberry Pi. Although it was not implemented, an ESP32 or ARM Cortex-M could also be used for this purpose. A third high-end approach, employs the Goertzel algorithm on a smartphone and is provided for reference purposes only. After all, detection with a smartphone (victim's device) is essentially no longer external detection, as it will not help to prevent the ultrasonic invasion. Nevertheless, this detector is useful to indicate whether the ultrasounds detected by one of the other approaches can be effectively used as a covert channel. In case the FOCUS implementation on the smartphone does not detect ultrasound, a malicious application on the smartphone probably cannot either and no privacy threat is present. Finally, a smartphone app is proposed that integrates the low-end and high-end approaches, alerting the user of the presence of malicious activity.

6.1 Low-end: Arduino

For this implementation, an Iduino 1485297 microphone volume sensor with an operating range of 50 Hz to 20 kHz is used. The digital output of this microphone is connected to the interrupt pin of an Arduino Uno Rev 3 that uses an ATmega328P microprocessor running at a clock frequency of 16 MHz. The sampling rate of the ADC of this Arduino model is 9600 Hz and is therefore insufficient to utilize Goertzel or FFT algorithms. The digital output of the sound sensor produces a high signal if the intensity of the detected sound is above a certain threshold and a low signal below this threshold. This threshold is adjustable with a potentiometer and, for this research, was experimentally set to the lowest possible value at which ambient noise does not produce a high signal. In this way, the highest possible sensitivity can be obtained. For the supply voltage we use 5V, which ensures that the high output is also 5V.

A program was subsequently written in the Arduino IDE that, using the Arduino's built-in hardware interrupts, counts the number of edge transitions of the microphone pulses over a 100 ms period. Each period of a sine wave causes four transitions from high to low or from low to high by intersecting the upper or lower threshold. From this, it can be concluded that if more than 6800 transitions are detected during a 100 ms period, sound with a frequency higher than 17 kHz is present. However, in order to account for deviations of pulses that are not detected (e.g., due to ambient noise), the lower threshold is set in practice at 6750 transitions. Because the sensitivity of the microphone for frequencies higher than 20 kHz is too low, no pulses will be generated for these frequencies and consequently an upper threshold is irrelevant. The pseudo-code of the algorithm for the digital counter software high-frequency detector is given in Algorithm 1.

Algorithm 1 Software high-frequency detector

```
1: if first_measurement then
2:    timestamp ← millis()
3:    first_measurement ← false
4: else
5:    if millis() − timestamp >= 100 then
6:       if number_of_pulses >= 6750 {17kHz} then
7:          detected ← true
8:       end if
9:       first_measurement ← true
10:      number_of_pulses ← 0
11:   end if
12: end if
```

6.2 Mid-end: Raspberry Pi

For the mid-end implementation, a Lioncast Universal USB microphone is used, connected to a Raspberry Pi model 3 v1.2. This Raspberry Pi runs on a quad-core Broadcom BCM2837 CPU with a clock speed of 1.2 GHz. The digital data measured by the microphone are read out in Python using the PyAudio library. A sampling rate of 44.1 kHz and a chunk size of 8192 are used for this purpose. Next, for each chunk, it is determined whether ultrasound is present. Depending on whether this sound is present or not, a message is displayed or the next chunk is taken in, respectively. The detection of ultrasound in a given chunk is done by applying the Goertzel algorithm[6] from Algorithm 2 to the chunk for each frequency between 17 and 20 kHz in steps of 10 Hz. Subsequently, the calculated magnitude is compared to a threshold to determine the presence of ultrasound signals. This threshold is empirically set at the lowest possible value at which no false positives are detected, which is 50'000. The variable "coeff" in the algorithm

[6] https://www.embedded.com/the-goertzel-algorithm/

is a constant determined by the sampling frequency, the target frequency and the chunk size.

Algorithm 2 Goertzel algorithm for 1 considered frequency

```
for every sample do
    Q0 ← coeff * Q1 − Q2 + sample
    Q2 ← Q1
    Q1 ← Q0
end for
magnitudeSquared ← Q1^2 + Q2^2 − Q1 * Q2 * coeff
```

6.3 High-end: smartphone

The high-end implementation uses the built-in microphone of a Samsung Galaxy S8+, which runs on an octa-core Exynos 8895 CPU with a clock speed of 2.3 GHz. The detection of ultrasonic signals is done in an analogous way for this implementation as for the mid-end approach. However, an application developed via Android Studio in Java is used here, rather than a Python script. The measured digital data from the microphone are read out in this application using the AudioRecord API at a sampling rate of 44.1 kHz. For the chunk size, we opt to take the minimum size supported by the device. Specifically for the Samsung Galaxy S8+, the chunk size is 3528. For this implementation, the threshold for the magnitude calculated with the Goertzel algorithm is set at 100'000.

When an ultrasound signal is detected, the International Mobile Equipment Identity (IMEI) number, phone number and software version of the device are automatically sent to a Cloud Firestore NoSQL database. The IMEI number, which usually consists of 15 digits, is a unique identifier for each cell phone. Since this number could, for example, be used to track a stolen device or to unlock a device, it could, coupled with the information from the ultrasonic signal, pose a serious privacy threat for the victim. In addition to sending personal data to the database, the FOCUS application alerts the user of the malicious activity by means of a notification, indicating the frequency at which ultrasound has been detected.

7 Evaluation

In this section, we discuss the performance evaluation of FOCUS in terms of detection distance and frequency sensitivity, based on our proof-of-concept implementation described in Section 6.

7.1 Detection distance

First of all, the maximum distance at which a reliable detection is possible is determined for the various detection methods. For this purpose, the transmitter uses a sine wave of 17 kHz emitted by the built-in speakers of a MacBook

Pro A1989 at different volume levels. These speakers are calibrated using the Audio/Musical Instrument Digital Interface (MIDI) settings to operate at 44.1 kHz, in order to match the sampling rate of the mid-end and high-end detection methods and in order to consider the lowest quality transmitter capable of generating UB signal. To generate the sine signals, the Szynalski[7] Online Tone Generator is used. For the mid-end and high-end FOCUS implementation, three different positions relative to the speakers are verified: above the laptop, next to the laptop, and in front of the laptop. Where the first position is more focused on detecting commercial advertisement tracking, the last two positions are more tailored to simulate a typical desk environment (e.g., for web tracking). For the low-end implementation, however, only the position above the laptop is considered in the measurements. This is because the other positions do not give a sufficiently reliable result, with in many cases even no detection possible. For each position, three volume levels are considered based on the MacBook Pro's volume settings: maximum volume, half volume, and quarter volume. Measured with the Decibel X app[8] directly above the speakers, for a frequency of 17 kHz, these levels correspond to 92 dB, 84 dB and 67 dB, respectively. For the high-end approach, an additional fourth level is considered, corresponding to a setting of 1/16 volume and 57 dB. All measurements are performed in a 450 cm by 350 cm room with a normal level of background noise (e.g., people talking). Due to the dimensions of the room, and because of their limited relevance, the exact distance is not determined for distances greater than 200 cm and less than 1 cm.

Since the low-end detection mechanism operates in a different manner than the mid-end and high-end mechanisms, different criteria are imposed to determine when reliable detection occurs. For the Arduino, we look at a status LED that turned on for 100 ms with each detection. If this LED remains permanently high, it can be concluded that the considered distance can be reliably detected. For the Raspberry Pi and the smartphone we look at whether detection is possible for the considered distance three times in a row within the three iterations of the Goertzel algorithm.

Table 1. Detectable distances with a 17 kHz signal

	low-end			mid-end			high-end		
	Position (cm)			Position (cm)			Position (cm)		
Volume level (dB)	1	2	3	1	2	3	1	2	3
95	40			>200	140	160	>200	>200	>200
84	30			160	25	90	>200	>200	>200
67	10			60	10	50	>200	>200	>200
57							>200	80	175

Table 1 shows the resulting distances for the different detection scenarios for a frequency of 17 kHz. One can see that all of the FOCUS implementations

[7] https:www.szynalski.comtone-generator

[8] https://play.google.com/store/apps/details?id=com.skypaw.decibel&hl=nl&gl=US

are capable of detecting the 17 kHz signal under every circumstance. Moreover, the high-end method can detect further than the mid-end approach and the latter further than the low-end approach. In addition, it can be stated that for position 1, further distances can be achieved than for positions 2 and 3. This can be explained by the orientation of the laptop speakers.

7.2 Frequency sensitivity

To verify the frequency sensitivity of the different FOCUS implementations, the different measurements are replicated with the same criteria at a frequency of 20 kHz. In this way, we consider both the lowest and the highest frequency of the UB, in order to allow for a reliable detection. The same volume levels are considered for this experiment, but for 20 kHz this correspond to 84 dB, 72 dB, 63 dB and 45 dB, respectively.

Table 2. Detectable distances with a 20 kHz signal

	low-end			mid-end			high-end		
	Position (cm)			Position (cm)			Position (cm)		
Volume level (dB)	1	2	3	1	2	3	1	2	3
84	60			>200	50	130	>200	>200	>200
72	20			50	< 1	10	>200	>200	>200
63	< 1			20	< 1	< 1	>200	80	110
45							35	5	10

Table 2 shows the resulting distances for the different detection scenarios for a frequency of 20 kHz. One can see that, again, all of the implementations are capable of detecting the signal under every circumstance. In general, it can be concluded that for a 20 kHz signal, lower detection distances can be realized than for a 17 kHz sine wave. This is because the sensitivity of a typical consumer microphone drops as the signal approaches 20 kHz. An exception to this observation is position 1 for the low-end detection method, where for the 20 kHz signal a greater distance was established. Possibly this can be explained by deviations in the signal caused by the acoustics, resulting in insufficient transitions being detected to reach the lower threshold.

7.3 Real-world performance

To demonstrate the real-world implications of FOCUS, 30 international websites were evaluated for ultrasonic beacons through FOCUS. Each of these websites was visited, using a VPN, from India and the USA. For each website, the scenario of an everyday user was assumed (e.g., adding an item to the shopping basket) and, where possible, a number of videos on the site were also evaluated. In this process, in one video [9], ultrasound was detected by both the smartphone

[9] https://www.instructure.com/en-au/canvas/resources/higher-education/canvas-learning-management-platform-across-globe-higher-education-students-leaders

and the Raspberry Pi implementation of FOCUS. However, due to the lack of data, it cannot be stated with certainty whether this is effectively a beacon. Moreover, as previously mentioned by [2], who monitored the audio output of the top 500 Alexa websites, finding beacons is a time-consuming process that can be compared to looking for a needle in a haystack. All this suggests that larger-scale research is needed.

8 Security Analysis

Our main goal is to detect ultrasonic sound in the victim's vicinity and inform the victim about the ultrasound detection. Although we can not prevent the ultrasonic invasion yet, detecting the signal and notifying the user will still help to raise awareness in order to take precautions.

For the three attack scenarios from the adversarial model, as described in section 5, it is known that commercial advertisement tracking is already actively employed by commercial companies (e.g., Shopkick, Lisnr, Signal360). Also, previous research has already shown that web tracking is possible by de-anonymizing a session in the Tor browser [14]. By intercepting the ultrasound signals between the attacker and the victim, FOCUS is able to detect both attack scenarios. In addition, FOCUS also demonstrates the danger of a data-theft scenario. In the high-end detection method, personal information is sent to a database upon detection. By implementing this approach across different apps with different privileges on different devices, it is possible to establish an aggregated profile of the victim. For this scenario as well, by intercepting the ultrasonic signals, FOCUS can demonstrate that a malicious activity is happening.

9 Discussion

In this article, our main objective is to detect UB signals and to alert the user of this clandestine practice. Particularly, we verify different possible detection methods against different positions and sound levels. Nevertheless, in addition to the advantages, each mechanism has specific disadvantages.

For the Arduino approach, it is not possible to determine at which exact frequency the detected sound was located, only whether ultrasound was present or not. Although the software high-frequency detector employed should theoretically be able to determine the frequency, it yields inconsistent results during measurements. In addition, for this approach, reliable detection is only possible in the extension of the speakers and the maximum detection distance is small compared to the other FOCUS implementations. However, since this implementation would be mounted directly against the speaker being monitored, this does not pose a threat to the effectiveness of this solution.

Another disadvantage of the low-end approach is the fact that detection is not possible under the presence of strong noise at lower audible frequencies. The main reason for this lies in the fact that the microphone used (as well as most consumer microphones) has a higher sensitivity for signals with a frequency

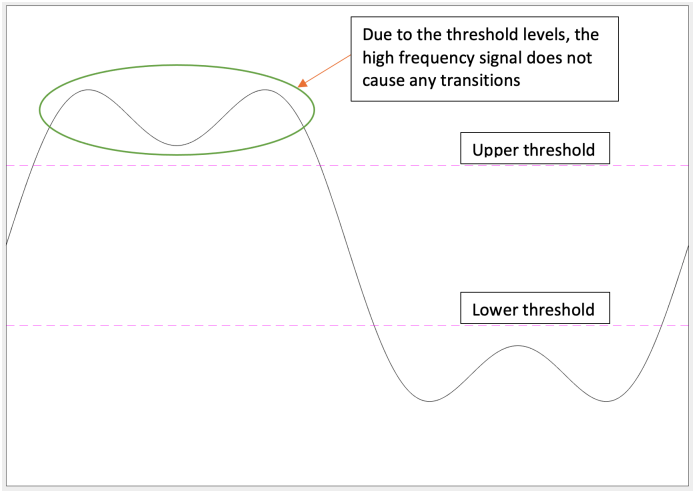

Fig. 3. Transition between high and low states of microphone volume sensor

lower than 17 kHz than for ultrasonic signals. As a result, the superposition of the audible signals and the ultrasonic signals does not produce sufficient transitions to reach the threshold of the digital counter. This is also shown in Figure 3, where only the signal with the lower frequency is able to cause transitions.

One solution, which was also verified in Simulation Program with Integrated Circuit Emphasis (SPICE), would be to use the analog output instead of the digital output of the microphone. This output can then be filtered using a high-pass filter (e.g., a second-order Butterworth high-pass filter) so that only the ultrasonic signals are passed through. By subsequently comparing the resulting signal to an adjustable reference voltage via an operational amplifier comparator, it is possible to detect whether ultrasonic beacons are present in the measured signal. To make this method less sensitive to high-frequency noise, the comparator can be a Schmitt trigger that provides hysteresis.

The smartphone FOCUS implementation on the other hand has the advantage that it can achieve the most accurate detection of the three approaches. However, this approach carries the disadvantage of high battery consumption on the detecting device, due to constant resource consumption of the Goertzel algorithm. For practical applications, it is therefore unlikely that it will be used for long periods of time. However, if it is known in advance which frequencies are being exploited by the attacker, the Goertzel algorithm can only be executed on these frequencies and low-power detection is possible.

As for the FOCUS smartphone application, we are aware that the Android Operating System requires permission to access the microphone and the phone, the former to detect the ultrasound and the latter to access the IMEI number and the phone number, which are sent to the Cloud Firestore Database upon detection. This paper assumes that the malicious app components are embedded

in an app for which the user would unsuspectingly accept these permissions (e.g., apps that enable phone calls).

Since FOCUS focusses on the presence or absence of a sine wave, it is capable of detecting both FSK and OOK modulation schemes. These are the only schemes used in practice, because PSK causes phase discontinuities resulting in audible clicks. In addition, as the Arduino implementation of FOCUS relies on the zero crossing rate, it could theoretically be generalized to detect DolphinAttack as well when paired with an ultrasonic microphone. Indeed, DolphinAttack is based on double-sideband transmitted-carrier amplitude modulation (AM-DSB-TC). By using the zero crossing rate, it should be possible to detect the carrier and therefore the presence of the attack signal.

10 Conclusion and Future Work

This paper presents FOCUS, an environment for the detection of ultrasonic invasion. We develop three different approaches for ultrasound detection: low-end, mid-end and high-end. The low-end approach employs a novel low-cost software high-frequency detector that utilizes a digital counter. While in the measurements, the high-end method serves as a reference, the mid-end and low-end approaches are compared in terms of detection distance and frequency sensitivity. We demonstrate the performance of FOCUS through two proof-of-concept implementations based on a Raspberry-Pi and an Arduino, respectively. The results confirm both the practicality and efficiency of FOCUS.

In future work we will, in addition to the detection, develop a means of preventing the attacks. Further, we will explore the practical applications of ultrasonic invasion (e.g., the exploitation of electronic voting machines).

11 Acknowledgement

This work is supported by the ESCALATE project, funded by FWO and SNSF (G0E0719N), and by Cybersecurity Initiative Flanders (VR20192203).

References

1. Arp, D., Quiring, E., Wressnegger, C., Rieck, K.: Privacy threats through ultrasonic side channels on mobile devices. In: 2017 IEEE European Symposium on Security and Privacy (EuroS P). pp. 35–47 (2017). https://doi.org/10.1109/EuroSP.2017.33
2. Arp, D., Quiring, E., Wressnegger, C., Rieck, K.: Bat in the Mobile: A Study on Ultrasonic Device Tracking. `https://www.sec.cs.tu-bs.de/pubs/2016-batmobile.pdf` (2016)
3. Arp, D., Quiring, E., Wressnegger, C., Rieck, K.: Privacy threats through ultrasonic side channels on mobile devices. In: 2017 IEEE European Symposium on Security and Privacy (EuroS& P). pp. 35–47 (2017). https://doi.org/10.1109/EuroSP.2017.33

4. Butler, C.: Visa makes strategic investment in LISNR, a start-up that wants to rival technology used by Apple Pay. https://www.cnbc.com/2019/11/05/visa-invests-in-lisnr-a-start-up-that-wants-to-rival-apple-pay.html (Nov 2019), [Accessed: 01.05.2021]
5. Carrara, B., Adams, C.: On acoustic covert channels between air-gapped systems. In: Cuppens, F., Garcia-Alfaro, J., Zincir Heywood, N., Fong, P.W.L. (eds.) Foundations and Practice of Security. pp. 3–16. Springer International Publishing, Cham (2015)
6. Deshotels, L.: Inaudible sound as a covert channel in mobile devices. In: 8th USENIX Workshop on Offensive Technologies (WOOT 14). USENIX Association, San Diego, CA (Aug 2014), https://www.usenix.org/conference/woot14/workshop-program/presentation/deshotels
7. Guri, M., Solewicz, Y., Elovici, Y.: Fansmitter: Acoustic data exfiltration from air-gapped computers via fans noise. Computers Security **91**, 101721 (2020). https://doi.org/https://doi.org/10.1016/j.cose.2020.101721, https://www.sciencedirect.com/science/article/pii/S0167404820300080
8. Guri, M., Solewicz, Y., Elovici, Y.: Speaker-to-speaker covert ultrasonic communication. Journal of Information Security and Applications **51**, 102458 (2020). https://doi.org/https://doi.org/10.1016/j.jisa.2020.102458, https://www.sciencedirect.com/science/article/pii/S2214212619304697
9. Guri, M., Solewicz, Y.A., Daidakulov, A., Elovici, Y.: Diskfiltration: Data exfiltration from speakerless air-gapped computers via covert hard drive noise. CoRR **abs/1608.03431** (2016), http://arxiv.org/abs/1608.03431
10. Joseph, T., Tyagi, K., Kumbhare, D.R.: Quantitative analysis of dtmf tone detection using dft, fft and goertzel algorithm. In: 2019 Global Conference for Advancement in Technology (GCAT). pp. 1–4 (2019). https://doi.org/10.1109/GCAT47503.2019.8978284
11. Lampson, B.W.: A note on the confinement problem. Commun. ACM **16**(10), 613–615 (Oct 1973). https://doi.org/10.1145/362375.362389, https://doi.org/10.1145/362375.362389
12. Madhavapeddy, A., Sharp, R., Scott, D., Tse, A.: Audio networking: the forgotten wireless technology. IEEE Pervasive Computing **4**(3), 55–60 (2005). https://doi.org/10.1109/MPRV.2005.50
13. Matyunin, N., Szefer, J., Katzenbeisser, S.: Zero-permission acoustic cross-device tracking. In: 2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST). pp. 25–32 (2018). https://doi.org/10.1109/HST.2018.8383887
14. Mavroudis, V., Hao, S., Fratantonio, Y., Maggi, F., Kruegel, C., Vigna, G.: On the privacy and security of the ultrasound ecosystem. In: Proceedings on Privacy Enhancing Technologies. pp. 95–112 (2017). https://doi.org/https://doi.org/10.1515/popets-2017-0018.
15. Mavroudis, V., Hao, S., Fratantonio, Y., Maggi, F., Kruegel, C., Vigna, G.: On the privacy and security of the ultrasound ecosystem. In: Proceedings of the 17th Privacy Enhancing Technologies Symposium. pp. 95–112. PETS '17, DE GRUYTER. https://doi.org/10.1515/popets-2017-0018