# Near-Ultrasound Inaudible Trojan (NUIT): Exploiting Your Speaker to Attack Your Microphone

Qi Xia
*Department of Electrical and Computer Engineering*
*University of Texas at San Antonio*

Qian Chen
*Department of Electrical and Computer Engineering*
*University of Texas at San Antonio*

Shouhuai Xu
*Department of Computer Science*
*University of Colorado Colorado Springs*

## Abstract

Voice Control Systems (VCSs) offer a convenient interface for issuing voice commands to smart devices. However, VCS security has yet to be adequately understood and addressed as evidenced by the presence of two classes of attacks: (i) *inaudible* attacks, which can be waged when the attacker and the victim are in proximity to each other; and (ii) *audible* attacks, which can be waged *remotely* by embedding attack signals into audios. In this paper, we introduce a new class of attacks, dubbed *near-ultrasound inaudible trojan* (NUIT). NUIT attacks achieve the best of the two classes of attacks mentioned above: they are *inaudible* and can be waged *remotely*. Moreover, NUIT attacks can achieve *end-to-end unnoticeability*, which is important but has not been paid due attention in the literature. Another feature of NUIT attacks is that they exploit victim speakers to attack victim microphones and their associated VCSs, meaning the attacker does not need to use any special speaker. We demonstrate the feasibility of NUIT attacks and propose an effective defense against them.

## 1 Introduction

Voice Control Systems (VCSs) are widely used in smart devices, especially those which do not have keyboards, including smartphones and smart home devices such as iPhone and Alexa. VCSs offer a great deal of convenience by allowing users or owners to use *voice commands* to activate and operate VCS devices, such as asking iPhone to make phone calls or send text messages when driving, or asking Alexa to play music or control other devices (e.g., smart home devices including locks). This is made possible by advancements in *speech recognition*, which uses artificial intelligence/machine learning (AI/ML) techniques to recognize voice commands.

Like any new technology, the security of VCS devices has yet to be thoroughly analyzed. A body of existing literature proposed the two classes of attacks discussed below.

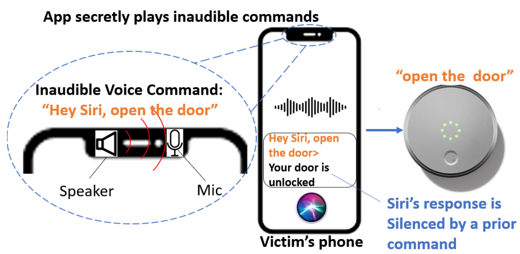
One class of attacks uses *inaudible* voice commands to attack VCS devices (e.g., smart phones) [1–4]. These attacks are stealthy because the attack signals are inaudible to humans but can be understood by VCS devices. For example, the DolphinAttack [2] and its siblings [1, 3] modulate *audible* voice commands into *inaudible* ultrasound signals, which are then used to attack VCS devices. These attacks exploit a physical property of VCS devices, known as *microphone nonlinearity*, which basically says that when the input signal's sound pressure level is high, a microphone can generate unexpected frequency components [1]. For technical reasons, these attacks can only be waged from a short distance between the attack device and the victim device, despite efforts at enlarging the distance [4]. In addition to ultrasound, inaudible attacks can also exploit laser technology [5].

Another class of attacks hides attack commands into some audible carrier audio (e.g., music). Two examples are CommanderSong [6] and Metaphor [7]. Unlike the preceding class of attacks, these attacks do not require the attacker-victim proximity assumption because they can be waged *remotely*, which will be referred to as *remote capability* hereafter. However, the requirement of audible base media (e.g. music) limits the attack to only non-silent attack scenarios, rendering these attacks noticeable by careful users especially when they are in a quiet environment.

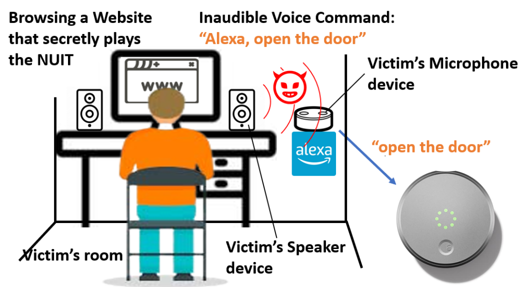
In this paper, we propose a *new* class of attacks, which modulate voice commands into *near-ultrasound inaudible* signals and embed these signals into an *appropriate* carrier (e.g. app, website or video); this is similar to embedding a Trojan Horse into an innocent program. We call the new family of attacks *near-ultrasound inaudible trojan* (NUIT).[1] When audio with embedded NUIT signals is replayed, the NUIT signals will attack a victim VCS device, which is also similar to how Trojan Horses are activated to wage attacks. From an attacker's point of view, NUIT attacks have three salient features. (i) They achieve the best of the two known classes of attacks mentioned above, by simultaneously entertaining *inaudibility* (as NUIT signals are inaudible) and *remote capability* (as the attacker can wage attacks remotely). (ii) They can achieve

[1] "Nuit" is a French word which means "night" in English.

*end-to-end unnoticeability*, which we define as *inaudible attack signals and silent responses*. This is important because the response of a smart device to an inaudible command may be audible and thus may alert the victim about the presence of attacks. (iii) They do not require the attacker to use any special hardware; instead, the attacker exploits victim speakers to attack victim microphones and their associated VCSs.




(a) Illustration of `NUIT-1`.




(b) Illustration of `NUIT-2`.

Figure 1: Illustration of two instances of the `NUIT` attack.

`NUIT` has two instances, which differ in whether the victim speaker and the victim microphone are on the same device or not. In the instance dubbed `NUIT-1` and illustrated in Figure 1a, the victim device runs an app, which secretly replays audio with embedded `NUIT` signals; as a consequence, the `NUIT` signals attack the microphone and the associated VCS on the *same* device to open a smart lock. In the instance dubbed `NUIT-2` and illustrated in Figure 1b, the victim uses a computer to browse a website, which replays audio with embedded `NUIT` signals to attack the microphone and Alexa on a *different* device to open a smart lock.

**Challenges in Realizing `NUIT` Attacks**. To wage `NUIT` attacks, we must tackle three challenges. The first challenge is to make the `NUIT` attacks (i.e., both `NUIT-1` and `NUIT-2`) able to exploit the limited bandwidth of Commercial-Off-The-Shelf (COTS) speakers to attack victim microphones and their associated VCSs. This challenge has no counterpart in previous *inaudible* attacks where the attacker uses special speakers; by contrast, `NUIT` exploits victims' COTS speakers. This challenge also has no counterpart in previous *remote* attacks because their attack signals are audible; by contrast, `NUIT` signals are inaudible. We address this challenge by using the Single-sideband Amplitude Modulation (SSB-AM) scheme [8, pp. 30], while adapting its demodulation method to leverage the microphone nonlinearity. It is worth mentioning that a windowed `NUIT` signal contains burst noise caused by spectral leakage; this can be addressed by leveraging the Tukey window [9] (cf. Appendix C).

The second challenge, which is relevant to the `NUIT-1` attack (but not the `NUIT-2` attack), is to embed `NUIT` signals into the limited time window imposed by the fact that VCS devices immediately mute, or lower the volume of, their speakers after processing the activation keyword (e.g., “Hey Siri” for Apple devices); this design is intended to make devices able to hear the subsequent action commands from the user clearly (e.g., “Open the door”) without interference from the device's own speaker. This matter is relevant because when the speaker is muted or turned down, it cannot be exploited to wage `NUIT-1` attacks. We address this challenge by identifying and exploiting the *reaction time window*.

The third challenge is to make the `NUIT` attacks (i.e., both `NUIT-1` and `NUIT-2`) achieve *end-to-end unnoticeability*, which we define as *inaudible attack signals and silent responses*. This is important because VCSs' responses to voice commands can be audible (e.g., Siri would respond to the *inaudible* command “open the door” with an *audible* response like “ok, the door is open”), thus alerting the victim about the presence of attacks. This issue is inherent to the *system design* of VCS devices, and does not appear to have been mentioned in the literature until very recently [10], where the authors suggest that the attacker may send an inaudible command (e.g. “turn the volume to 3”) to turn down the victim device's speaker to an inaudible level to make the VCS' response unnoticeable. This method can be applied to make `NUIT-2` achieve end-to-end unnoticeability. However, this method fails to make `NUIT-1` achieve end-to-end unnoticeability because `NUIT-1` exploits a victim's speaker to attack the same victim's microphone and VCS on the same device; for many VCS devices, turning down their speaker also makes `NUIT-1` fail. We address this challenge by testing VCSs' response mechanism to find that `NUIT-1` can attack Siri devices while achieving end-to-end unnoticeability.

**Our Contributions**. We make four contributions. ***First***, we introduce a new class of attacks against VCS devices, dubbed `NUIT`, which can simultaneously achieve the *inaudibility* of attack signals, the *remote capability* for waging attacks, and the *silent response* as devices permit. `NUIT` has two instances: `NUIT-1` exploits a victim's speaker to attack the same victim's microphone and VCS on the *same* device; `NUIT-2` exploits a victim's speaker to attack the same victim's microphone and VCS on a *different* device. ***Second***, we demonstrate the feasibility of `NUIT`, by addressing the three challenges mentioned above. The ideas we use to address these challenges may be of independent value, such as the adaptation of the SSB-AM modulation to achieve inaudibility. Mathematical reasoning of SSB-AM demodulation to leverage the microphone nonlinearity. To help understand `NUIT`, we make our attack demo videos available at [11]. ***Third,*** we find that the `NUIT` attacks fail to attack iPhone 6 Plus, which reminds us that the DolphinAttack also fails to attack iPhone 6 Plus [2]. Since there is no explanation for why iPhone 6 Plus can resist these attacks, we conduct a study and find the reason is that its microphone has weak nonlinearity, which is caused by its low gain audio amplifier. This does not means that using microphones with weak nonlinearity is a good strategy to harden the security of devices, because it also hurts the legit use of VCSs.

Fourth, since known defenses have limitations in defending against NUIT, we propose a single-factor software-based defense, which leverages the attack's success to counter it, as follows: When the attack succeeds, the victim microphone must have detected and recognized the embedded NUIT signals at a near-ultrasound frequency; this capability can be leveraged to detect NUIT. We use simulation to evaluate our defense because the VCS devices available to us do not have open-source code or interfaces we can use. Simulation results show that it has zero false-positives and zero false-negatives, which is attributed to the leverage of physical properties of VCS devices.

Other Scenarios of NUIT Attacks. There are many ways to wage NUIT attacks than what is illustrated in Figure 1, such as the following. (i) NUIT can be waged in a standalone fashion— the attacker uses its own COTS speaker to attack a victim's microphone and VCS, as in the case of the DolphinAttack [2] and its siblings. (ii) Figure 1b illustrates that the NUIT-2 attacker can exploit victim A's speaker on one device to attack A's microphone on another device. This attacker could exploit A's speaker to attack B's microphone, for example when A and B sit next to each other.

Ethical Issues Since NUIT exploits physical properties of COTS speakers and microphones, rather than software vulnerabilities, spreading awareness is a sensitive matter. This is similar to what was encountered by the DolphinAttack [2] and its siblings [5, 10, 12]. Nevertheless, our attack experiments are conducted in controlled environments against our own devices and pose no threats to others.

Paper Outline. Section 2 reviews related prior studies. Section 3 describes preliminary knowledge. Section 4 discusses the threat model. Section 5 addresses the challenges to realizing the attacks. Section 6 demonstrates the feasibility of NUIT. Section 7 analyzes the factors affecting the success of NUIT. Section 8 investigates defense against NUIT. Section 9 discusses the limitations of the study. Section 10 concludes the paper. Some details are deferred to Appendices.

## 2 Related Work

Prior Studies Related to Near-Ultrasound Signals. In the literature, near-ultrasound signals have been used to synchronize TV shows with smart device app services [13], facilitate two-factor authentication [14], and enable wearable medical devices communications [15], medium-range (25m) communications [16], and high-throughput communications between COTS devices [17]. By contrast, NUIT is the first to exploit near-ultrasound signals to wage attacks against VCS devices. Table 1 compares these studies, highlighting their differences in modulation scheme (details can be found in the respective papers), communication distance, data rate, and whether to exploit microphone nonlinearity (mic NL for short) or not.

Table 1: Comparing studies related to near-ultrasound signals.

| Reference | Modulation Scheme | Maximum Distance | Data Rate (kbps) | mic NL? |
|---|---|---|---|---|
| 2ndScreen [13] | QOK | 2.7m | >15 | No |
| UWear [15] | OFDM/GMSK | N/A | 2.76 | No |
| Chirp-based [16] | Chirp | 25m | 16 | No |
| Batcomm [17] | OFDM+DSB-AM | 10cm | 47 | Yes |
| NUIT | SSB-AM | 4.6m | N/A | Yes |

Prior Studies on Attacks Related to NUIT. As mentioned above, we divide previous attacks related to NUIT into two classes: inaudible vs. audible. Table 2 compares previous attacks and NUIT. Previous inaudible attacks carry attack signals via electromagnetic waves [18, 19], laser beams [5], or ultrasound waves [1, 2, 4, 10, 12] (through air while assuming line-of-sight or LOS [1, 2, 4], or through solid material [10]). Previous audible attacks are incomprehensible to humans [6, 7, 20, 21]. But attacks in [20, 21] sound like random noises to humans and may alert the presence of attacks. CommanderSong [6] and Metaphoer [7] require audio (e.g. music) to hide the command, thus cannot achieve inaudibility. These attacks exploit either the difference in computer vs. human speech recognition systems [20], or adversarial examples against computer speech recognition systems [6, 7, 21].

Among the attacks reviewed above, CommanderSong [6] is closely related to NUIT because they both can be waged remotely by embedding attack signals into some audible carrier media (e.g., video/audio). However, two differences make NUIT more stealthy. (i) NUIT is not noticeable to the victim user even in a quiet environment, owing to the use of inaudible attack signals by design; whereas, CommanderSong attack signals are audible noise-like signals by design. (ii) NUIT can embed inaudible attack signals into a silent app or website, but CommanderSong must use audible carrier media (e.g music).

Prior Studies on Defenses Related to NUIT. Known defenses against inaudible attacks can be divided into two categories: Single-factor defenses [2, 4, 10, 22] and Multi-factor defenses [22–25]. Single-factor defenses can further be divided into two sub-categories: hardware-based [22] vs. software-based [2, 4, 10]. Hardware-based Single-factor defenses (e.g. [22]) have the limitation that they require modification of device hardware, therefore fail to protect existing devices on the market that don't allow hardware modification. Software-based Single-factor defenses [2, 4, 10] detect "abnormal" behaviors in the frequency domain of commands received from the mono microphone to detect attack signals, which can be easily implemented on all existing devices via a software update; our defense belongs to this type. However, as elaborated in Appendix D, existing defenses can be evaded by a specially crafted attack signal (e.g. our SSB-AM based NUIT signal). Instead of just using the mono microphone, multi-factor defenses exploit additional sensors on certain VCS devices, (e.g. motion sensors [23], microphone array [24, 25], extra speakers [26]) to extract features in other dimensions to detect whether the received command is legit or not. These multi-factor defenses can defeat inaudible attacks including NUIT, but have the limitation that the victim VCS device must contain such additional sensors, and thus fail to protect most existing devices without such sensors.

Table 2: Comparison between previous attacks and NUIT where `R' means Range, `AF' means Attack Frequency, `LOS' denotes whether the attack requires line-of-sight (LOS) or not, `ST' means Special Transducer.

| Reference | R (m) | AF (Hz) | LOS | ST |
|---|---|---|---|---|
| Attacker exploits inaudible attack signals (e.g., ultrasound, laser) | | | | |
| Dolphin [2] | <1.75 | 20k | Yes | Yes |
| Long Range [4] | <11.89 | 20k | Yes | Yes |
| Backdoor [1] | <11.89 | 20k | Yes | Yes |
| Surfing [10] | N/A | 20k | No | Yes |
| Laser [5] | >100 | < 6k | Yes | Yes |
| CapSpeaker [12] | 0.105 | 20k | No | Yes |
| IEMI [18] | 1.2 | < 6k | No | Yes |
| Whisper [19] | Cable length | < 6k | No | Yes |
| NUIT (This work) | Remote | 16k-22k | No | No |
| Attacker embeds audible but human-incomprehensible attack signals into audible base audios (e.g. music) | | | | |
| CommanderSong [6] | Remote | <16k | No | No |
| Metaphor [7] | Remote | <6k | No | No |
| Attacker exploits audible but human-incomprehensible attack signals without using any carrier audios | | | | |
| CocainNoodle [20] | Remote | <6k | No | No |
| Hidden Voice [21] | Remote | <6k | No | No |

## 3 Preliminaries

**VCS User-to-Device Authentication.** A VCS has two main components. The *voice-capturing* component is responsible for capturing sound waves and digitizing them for further processing. This component consists of a microphone, an amplifier, a Low-Pass Filter (LPF), and an analog-to-digital converter (ADC), where LPF often operates at the frequency of 20kHz. The *speech recognition* component uses AI/ML to detect a device-specific *activation keyword* (e.g., “Hey Siri" for Apple, “Alexa” for Amazon, “Hey Google” for Google Assistants, and “Cortana” for Microsoft) and subsequent *action commands* (e.g., “Call phone #123-4567"). A VCS constantly listens for its activation keyword. We use the term *voice commands* to accommodate both activation keywords and action commands. A VCS uses voiceprint to authenticate the activation keyword, but we are not aware of any VCS device that uses voiceprint to authenticate action commands.

**VCS Response Mechanism and Its Implications.** VCSs often respond to action commands with confirmations, which appear to depend on their comprehension of an action command. For example, Siri would respond to the command "Open the door" with a response "Your door is open". Since the response to an inaudible action command may alert the presence of attacks, the attacker would want to silence the response. We find that Siri's responses are controlled by a separate mechanism rather than using the media volume, which makes it possible to achieve silent responses and end-to-end unnoticeability. However, Google Assistant, Cortana, and Alexa's responses use the same volume as their media volume, meaning that the attacker cannot silence responses without jeopardizing the success of NUIT attacks.

**Audible Frequency Range.** Human ears are most sensitive to sound with a frequency between 2kHz and 5kHz and insensitive to sound with a frequency higher than 16kHz [27, 28]. Sound with a frequency ≥ 16kHz is deemed *high frequency* to humans [17]. In this paper, the attacker modulates human voice commands in the frequency range 50Hz-6kHz [29] to sound waves at the *inaudible near-ultrasound frequency* between 16kHz and 22kHz.

**Double Sideband and Amplitude Modulation (DSB-AM) Is Not Sufficient for NUIT.** COTS speakers have a Digital-to-Analog Converter (DAC) with at least a sample rate of 44.1kSa/s (Samples per second). According to the Nyquist–Shannon Sampling Theorem [30], this means that the audio output frequency of COTS speakers is upper bounded at 22kHz. Since the minimum inaudible frequency is 16kHz, the frequency range of COTS speakers that can be used to wage inaudible attacks is 6kHz (i.e., 16kHz-22kHz), which is the range that can be exploited in theory. This is confirmed by our experiments as shown in Appendix A.

However, this 6kHz (i.e., 16kHz-22kHz) inaudible bandwidth is too narrow for the DSB-AM modulation scheme, which is used by previous inaudible attacks. This is because DSB-AM signals require at least 12kHz bandwidth (see Appendix B for details), which cannot fit into the 6kHz inaudible bandwidth of COTS speakers without causing audio leakage at the left sideband (i.e., frequency range 10kHz-16kHz), making the attack audible as shown in Figure 2. This means NUIT needs a different modulation scheme to accommodate the 6kHz inaudible bandwidth of COTS speakers.

Figure 2: Illustrating why DSB-AM cannot be used in NUIT.

# 4 Threat Model

The attacker's goal is to remotely exploit the speaker on a victim device to inject voice commands as NUIT into the microphone and associated VCS on the same device (NUIT-1) or on a different device (NUIT-2), without the victim user's notice during the delivery, invocation and execution of the attack. To achieve end-to-end unnoticeability, we assume no user interaction with the microphone device when NUIT is waged, otherwise victims may be alerted by the presence of attacks. For example, NUIT-1 can be waged by a malicious app running in the background when the victim is sleeping. Similarly, the microphone device is assumed not in use (regardless of the speaker device) when waging NUIT-2. The following requirements must be achieved for waging NUIT.

Phase 1. Stealthy Preparation. The attacker can embed NUIT signals into some appropriate carrier without being noticed. For example, the attacker can write a malicious app or compromise an innocent app that can replay a NUIT audio, or upload NUIT audio to social media platforms (e.g., YouTube). Moreover, the attacker has a sample (or adversarial example) of a victim user's activation keyword when voiceprint-based authentication is enforced. This is not difficult to achieve, as assumed in previous attacks.

Phase 2. Remote Delivery. We assume that the attacker can remotely deliver NUIT audio to a victim. For example, exploiting social engineering means luring a victim to download and install a malicious app that can replay malicious audio, or victims visit a malicious website as mentioned above.

Phase 3. Inaudible Invocation. NUIT attacks can be invoked inaudibly when (i) the downloaded maliciously app is automatically replaying a silent audio in the background (or opened by the victim) and/or the maliciously website containing NUIT signals replaying a silent audio is visited by victims. This silent setting contains no carrier audio noise, which has never been achieved in previous studies. NUIT can also be automatically waged when victims are (ii) watching malicious videos that contain carrier audio noise, which is similar to the threat model of CommanderSong [6].

Phase 4. Unnoticeable Execution. The execution of the NUIT attack achieves end-to-end unnoticeability, meaning that the NUIT signals are inaudible and VCS responses are silent.

# 5 Addressing the Challenges

## 5.1 Addressing Challenge 1

One approach to addressing this challenge, namely making NUIT able to exploit the 6kHz bandwidth of COTS speakers, is to proceed in two steps. (i) Identify the minimum bandwidth that can be used to activate victim VCSs. (ii) Modulate voice commands into the inaudible frequency range of victim COTS speakers while assuring successful demodulation.

### 5.1.1 Identifying the Minimum Activation Bandwidth

To make NUIT widely applicable, we consider four popular VCS devices [31]: Amazon Alexa, Apple Siri, Google Assistant, and Microsoft Cortana. To accommodate them simultaneously, we identify the minimum bandwidth that is needed to activate them. For this purpose, we analyze their spectrum by repeatedly replaying their activation keywords and increasing the sample rate until they are activated. For example, we replay "Hey Siri" starting at a sample rate of 8kSa/s (i.e., 8k samples per second); if Siri is not activated, we try 12kSa/s, 16kSa/s, and so on, until Siri is activated. Experimental results show: Amazon Alexa, Google Assistant, and Cortana all require a sample rate of 8kSa/s for activation, but Siri requires a sample rate of 12kSa/s. Thus, making NUIT applicable to all these devices requires a minimum of 12kSa/s baseband sample rate (i.e., 6kHz baseband bandwidth [30]).

### 5.1.2 SSB-AM: Leveraging Microphone Nonlinearity to Cope with COTS Speaker Bandwidth Constraint

The attacker can use Single-Sideband Modulation-Amplitude Modulation (SSB-AM) [8, pp. 124–132] to modulate voice commands into the 6kHz bandwidth identified above.

SSB-AM Modulation. We briefly review the basic ideas while please refer to [8, pp. 125–129] for derivation details. The two forms of SSB-AM, namely the Upper Sideband Amplitude Modulation (USB-AM) signal, denoted by $S_{USBAM}$, and the Lower Sideband Amplitude Modulation (LSB-AM) signal, denoted by $S_{LSBAM}$, can be expressed as:

$$S_{USBAM}(t) = (1+v(t))\cos(2\pi f_c^u t) - \hat{v}(t)\sin(2\pi f_c^u t); \quad (1)$$
$$S_{LSBAM}(t) = (1+v(t))\cos(2\pi f_c^l t) + \hat{v}(t)\sin(2\pi f_c^l t); \quad (2)$$

where $v(t)$ is the baseband voice command signal and $\hat{v}(t)$ is its Hilbert transform [8, pp. 82–83], and $f_c^u$ and $f_c^l$ respectively denote the carrier frequency for $S_{USBAM}$ and $S_{LSBAM}$.

Now the question is: Should the attacker choose USB-AM or LSB-AM to modulate voice commands? To make NUIT inaudible, the attacker must assure that the spectrum magnitude is always below the threshold of the human hearing curve, which is illustrated in Figure 3. In theory, LSB-AM allows the attacker to set the carrier in the ultrasound frequency range ($>$ 19kHz) to generate high-power NUIT signals (up to 80db SPL), while making NUIT inaudible. In practice, however, many COTS speakers have increasingly deteriorated frequency responses going beyond 19kHz (see Appendix A). This means that using LSB-AM would lead to a low attack success rate for mobile devices. Although this can be compensated by using a high-volume speaker, it does not apply to most mobile devices. Thus, the attacker would use USB-AM with carrier wave at frequency $f_c^u = 16$kHz for most devices.

SSB-AM Demodulation. Now we discuss how SSB-AM modulated NUIT signals can be demodulated by COTS microphones. We focus on the demodulation of USB-AM signals, while noting that the idea equally applies to LSB-AM.

Figure 3: Illustrating the hearing curve and how to make NUIT signals inaudible for USB-AM and LSB-AM modulation.

Figure 4: Illustration of SSB-AM demodulation.

Figure 4 illustrates the basic idea. When a microphone receives the USB-AM signal $S_{USBAM}(t)$ given by Eq. (1), it generates the following output signal:

$$S_{out} = S_{USBAM}(t) + S^2_{USBAM}(t); \qquad (3)$$

where $S_{USBAM}(t)$ does not contribute to the attack because its frequency is above 16kHz (i.e., it is out of the speech frequency range and thus ignored by the VCS). [But, this linear term can be leveraged for defense as we will show later!] Note that the quadratic term $S^2_{USBAM}(t)$ has three components: a high-frequency $2f_c^u$ component

$$(v(t)+1)\hat{v}(t)\sin(2\pi 2f_c^u t) + \frac{v^2(t)+2v(t)+1-\hat{v}^2(t)}{2}\cos(2\pi 2f_c^u t);$$

a Direct Current (DC) component $1/2$, and an audible component $s_b(t) = \frac{1}{2}(v^2(t)+2v(t)+\hat{v}^2(t))$. The high-frequency component is filtered by the Low-Pass Filter (LPF) of the microphone with a cut-off frequency of 20kHz because $2f_c^u = 32kHz > 20kHz$. The DC component is filtered by the microphone's capacitor. Thus, only the audible component $s_b(t)$ and the linear component $S_{USBAM}(t)$ can pass the microphone filtering system. Moreover, only $s_b(t)$ contributes to the attack because $s_b(t)$ contains the voice command signal $v(t)$.

**Insight 1** COTS microphones are not designed to demodulate SSB-AM signals, but their nonlinearity happens to enable it.

## 5.2 Addressing Challenge 2

**Understanding and Measuring the Reaction Time** The concept of reaction time is inherent to all VCS devices. Upon receiving the activation keyword, VCSs either mute their speakers or lower their speakers' volume to its minimum. The reaction time is the interval between (i) when the activation keyword is received and (ii) when the speaker is muted or its volume is lowered. The reaction time is inevitable as it takes time for VCSs to process the activation keyword. The design—muting, or lowering the volume of, speakers after hearing the activation keyword—is for making the microphone listen to action commands without interference from the audio that is replayed by the speaker. Because (i) VCS can only mute, or lower the volume of, the speaker on the same device, and (ii) NUIT exploits victim speakers to wage attacks, the reaction time has one subtle yet important implication for NUIT-1, which exploits the speaker to attack the microphone on the same device, but not for NUIT-2 that exploits the speaker to attack the microphone on a different device.

Figure 5: Illustration of the injection of malicious action commands within the reaction time window in the NUIT-1 attack.

For the VCSs that mute the speaker after the reaction time, the attack cannot continue to exploit the muted speaker. Thus, the attacker's malicious voice commands must fit into the reaction time window; otherwise, the attack will fail. For the VCSs that lower the volume of the speaker after the reaction time, the attack can continue to exploit the speaker but may still fail (depending on the volume). To make NUIT-1 widely applicable, we propose always embedding action commands into the reactive time window, regardless of whether the speaker will be muted by the VCS, as illustrated in Figure 5. This explains why the reaction time imposes a hard constraint on NUIT-1, but not NUIT-2.

Table 3: Empirical reaction time of VCS devices.

| VCS | Reaction Time (sec) | Mute Speaker? |
|---|---|---|
| Siri | 0.82 - 1.53 | Yes |
| Google | 0.77 - 0.96 | Yes |
| Alexa | 0.79 - 0.94 | No |
| Cortana | 0.87 - 0.99 | No |

Table 3 summarizes the minimum and maximum reaction time observed among the 100 experiments we conducted with each device. The minimum reaction time is 0.77 seconds.

**Insight 2** To wage successful NUIT-1 attacks against Siri, Google Assistant, Alexa and Cortana devices, malicious action commands must not be longer than 0.77 seconds.

**Exploiting the Reaction Time.** In our experiment, we consider the action commands listed in Table 4 within the reaction time window of 0.77 seconds. These commands are useful to the attacker. Experimental results show that NUIT-1 successfully injects all these commands within 0.77 seconds.

**Insight 3** Many action commands can indeed fit into the reaction time window to wage the NUIT-1 attack.

Table 4: Action commands successfully injected by NUIT-1.

| Device (VCS) | Action Command |
|---|---|
| iPhone (Siri)<br>Echo Dot (Alexa)<br>Android Phone (Google Assistant)<br>Windows PC (Cortana) | -Speak 6%/Turn down volume<br>-Open the door/YouTube<br>-What's the time/day/weather<br>-Tell me a joke<br>-Read my message<br>-Call Sam<br>-Turn on light/airplane mode |

## 5.3 Addressing Challenge 3

Surfing attack [10] proposes sending inaudible action commands to reduce Google Assistant's response volume to Level 3 to prevent the response from being heard by the user before proceeding with further attack. NUIT-2 attack can directly adopt this method by first sending an action command "Turn volume to 6%" to the target microphone device to make the VCSs' response unnoticeable, and then proceed with subsequent attacks. Such method cannot be adopted by NUIT-1 because for many VCS devices (e.g. Google Assistant, Cortana, Alexa), lowering system volume also lowers NUIT-1 signal's volume, making further attacks impossible.

Nevertheless, we found that Siri is an exception. Our investigation shows that for iPhone Siri devices, the volume of the response and the volume of the media are separately controlled. Thus, the attacker can use an action command to mute Siri's response without muting the subsequent NUIT-1 commands. A running example of the NUIT-1 Attack muting Siri's response is detailedly described in Section 6.1.

Insight 4 For NUIT-1 attacks, only Siri's response can be silenced to achieve an unnoticeable attack but not the others.

# 6 The NUIT Attack

How to Embed NUIT into Carriers? We mentioned that NUIT signals need to be embedded into appropriate carriers (e.g. app, website, videos). Based on carrier audio's audibility, the embedding strategies are different: (i) The carrier audio itself is silent (i.e., blank or void), in which case NUIT signals can be embedded anywhere in the carrier audio. Examples of such carriers are apps and websites. (ii) The carrier audio is audible but contains some silent segments that are silent, dubbed silent segments for short, such as pauses in a speech and intervals between music soundtracks. In this case, NUIT signals should be embedded in the silent segments (otherwise, the attack might fail because the NUIT signals will be overwhelmed by the carrier audio). There are many ways to identify such silent segments in given audio, such as appending such segments to the end. Since it is popular to edit and share self-made audios, which may be associated with videos, on social network platforms, this would be one effective method for waging the NUIT attack. Examples of such carriers are YouTube videos. Note that the preceding attack scenario (i) does not have a counterpart in the CommanderSong attack [6] which uses audible carrier media, but (ii) is indeed similar to the CommanderSong attack because both use audible carrier media.

## 6.1 The NUIT-1 Attack

How Does the NUIT-1 Attack Work? At a high level, the attacker uses SSB-AM to modulate the activation keyword and malicious action command(s) into near-ultrasound signals, and then embeds these signals into some appropriate carrier audio to obtain malicious audio, which executes the attack when replayed. Details follow.

Phase 1: Preparation. This phase has four steps. (i) The attacker needs to understand the target VCS devices, including their reaction time and their response mechanism. (ii) The attacker needs to assure that the activation keyword can pass the voiceprint authentication of the target VCS devices that enforce it (e.g., Siri). This is readily doable [2], while noting that this is not needed for action commands because VCS devices do not authenticate them. (iii) The attacker needs to accommodate the limited bandwidth of COTS speakers, assure inaudibility when modulating voice commands, assure the voice commands can fit into a single reaction time window for all the VCS devices, and assure a silent response. This can be achieved by addressing Challenges 1-3 as shown above. This leads to NUIT signals. (v) The attacker embeds NUIT signals into some appropriate carrier audio as mentioned above, leading to malicious audio with embedded NUIT signals.

Phase 2: Delivery. The attacker uses social engineering to lure users to install the malicious app, visit the malicious website, or listen to the malicious audio.

Phases 3 and 4: Invocation and Execution. When a user runs a malicious app, visits a malicious website, or watches malicious videos, NUIT signals can attack the microphone on the same device in an end-to-end unnoticeable fashion.

A Running Example of NUIT-1 Attacking Siri.

Phase 1: Preparation. (i) The attacker needs to know that iPhone has two different volume controls for the response and the media. (ii) This is assured in our own attack experiment because we attack our own devices. (iii) In our attack experiment, we use two example action commands that can fit into a single action time window: one is "speak 6%" for lowering Siri's response volume to 6% to achieve end-to-end unnoticeability, and the other is "open the door" as the attack payload. (iv) In our attack experiment, we use Matlab code, which is our implementation of the SSB-AM modulation scheme, to generate the near-ultrasound signals of the activation keyword and the two action commands. This leads to two separate wav files, one for each action command (following the activation keyword). (v) In our attack experiment, we embed the NUIT signal, namely the wav file into two carriers: one is with silent audio (e.g. mobile app), in which case we embed it at an arbitrary place; the other is normal audio of music, in which case we append the wav file to the end of the audio. This leads to four wav files of malicious audio as there are two action commands and two carrier audios.

Phases 2-4: Delivery, Invocation, and Execution. In our attack experiment, we replay each of the four malicious audios to attack our own iPhone XR for ethical reasons. We observe that the iPhone XR device executes the "open the door" command with end-to-end unnoticeability as shown in the demo video we post on the website.

## 6.2 The NUIT-2 Attack

How Does the NUIT-2 Attack Work? In this case, the attacker exploits the speaker on one device of the victim to attack the microphone and associated VCS on another device of the victim. The attack is similar to NUIT-1, except for the following. The attacker does not need to deal with the reaction time (Challenge 2) and the response mechanism because they have no effect on NUIT-2 (Challenge 3). The reaction time has no effect because the first device's speaker will not be muted by the second device, assuming that the victim speaker device uses no VCS or a different VCS than the VCS used by the victim microphone device (i.e., an attack targeting Siri does not affect Alexa as their activation keywords are different).

A Running Example of NUIT-2 Exploiting iMac to Attack Google Assistant. In our attack experiment, the victim's first (speaker) device is an iMac 2020 desktop and the second (target) device is an Android LG ThinkQ smartphone using Google Assistant, while noting that NUIT-2 targeting Google Assistant cannot compromise iMac. Since the phases of NUIT-2 are similar to that of NUIT-1, we only highlight the differences between them. In NUIT-2, the attacker has more freedom in choosing action commands because the reaction time has no effect. We use two similar commands to attack Google Assistant, namely "turn the volume to 1" and "open the door." The carrier audio is silent. We embed the malicious audio into a webpage on our own iMac computer, which cannot be accessed from any other computer (for ethical reasons). When using the Chrome browser to visit this webpage, the Android LG ThinkQ indeed opens a smart lock.

## 6.3 Devices Vulnerable to NUIT Attacks

Table 5 summarizes the tested devices according to our experiments. We make the following observations. First, Apple iPhone X, XR and 8 are vulnerable to both NUIT-1 and NUIT-2 with end-to-end unnoticeability. Second, some devices are not vulnerable to NUIT-1. This can be attributed to (i) the distance between the victim speaker and the victim microphone, even on the same device, being too long to make the attack succeed, and/or (ii) the speaker quality on the victim device is not good enough. Third, some devices cannot be attacked by NUIT-1 or NUIT-2 with end-to-end unnoticeability because the attack cannot silence these devices' audible responses. Fourth, NUIT-1 and NUIT-2 fail to attack iPhone 6 plus. Note that the DolphinAttack also fails to attack iPhone 6 Plus [2], and the cause is not known. This prompts us to investigate the cause of this phenomenon below.

Table 5: Devices vulnerable to NUIT, where X means an attack succeeds with end-to-end unnoticeability, X* means an attack succeeds with inaudible attack signals but not silent response, and means an attack fails.

| Target VCS Device | NUIT-1 | NUIT-2 |
|---|---|---|
| iPhone: X, XR, 8 | X | X |
| MacBook: Pro-2021, Air-2017 | X* | X |
| Galaxy: S8, S9, A10e | X* | X |
| Echo Dot Gen1 | X* | X |
| Dell Inspiron 15 | X* | X* |
| Apple Watch 3 | | X |
| Google Pixel 3 | | X |
| Galaxy Tab S4 | | X |
| LG Think Q V35 | | X |
| Google Home 1 | | X |
| Google Home 2 | | X |
| iPhone 6 plus | | |

Why Does NUIT Fail to Attack iPhone 6 Plus? It is known that the nonlinear component in a microphone system is the amplifier [4]. This hints that NUIT (and DolphinAttack when waging common attack signals [2]) fail to attack iPhone 6 Plus because it has a low-gain amplifier, which has a weak nonlinearity that cannot be exploited to wage these inaudible attacks. To see this, let's recall that generally speaking, when the input voltage increases, the output voltage of an amplifier does not increase beyond a cutoff voltage, known as the saturation voltage denoted by $V_{sat}$. Moreover, the output is linear to the input signal when the output voltage is small, but does behave nonlinearly when the output voltage gets close to $V_{sat}$. This nonlinear region is exploited by DolphinAttack and NUIT to wage inaudible attacks. We suspect that these attacks are successful against devices including iPhone X, XR, and 8 because these devices use a high-gain amplifier, and that these attacks fail to attack iPhone 6 Plus because it uses a low-gain amplifier, which makes it hard to exploit the nonlinear region to make the attacks succeed. This is plausible because when the input is at a common level, a low-gain amplifier usually generates a small output voltage, which is far below $V_{sat}$ and thus makes the output linear to the input.

To validate the preceding discussion, we conduct experiments to compare the amplifier transfer curve of iPhone 6 Plus and iPhone X. The experiments are conducted by using a Vifa speaker [32] to send 18kHz sinusoidal acoustic signals at different decibel levels to the front microphone of both phones and analyzing their output voltage in the recorded files. For each phone, we send input sound pressure level (SPL) from 60 dB to 130 dB with an interval of 5dB, and record the output maximum voltage for each input. Figure 6 depicts the results, where the $x$-axis is the input 18kHz signal sound in a specific decibel, and the $y$-axis is the output voltage in

Figure 6: Microphone amplifier transfer curves of iPhone 6 Plus and iPhone X.

decibels with $V_{sat}$ normalized to 0dB. We observe that iPhone X has a high-gain amplifier with a nonlinear region starting at 73dB, whereas the output of iPhone 6 Plus is linear until reaching 115dB. This explains why a common decibel range ultrasonic signal (75dB-80dB) can successfully attack iPhone X but not iPhone 6 Plus. Moreover, the nonlinear region of the low-gain iPhone 6 Plus amplifier cannot be exploited unless the input reaches or goes above 115dB. This justifies the experiments in DolphinAttack [2] that iPhone 6 Plus can still be successfully attacked after placing the attacker speaker at a 2cm distance from the victim device when raising the attack signals to 125dB.

Table 6: Comparison of microphone sensitivity between three devices: iPhone 6 Plus, iPhone XR, and iPhone X, at various distances: from 5 cm to 50 cm. `Act.' stands for activation rate and `Rec.' stands for recognition rate.

| Distance | iPhone 6 Plus | | iPhone XR | | iPhone X | |
|---|---|---|---|---|---|---|
| | Act. (%) | Rec. (%) | Act. (%) | Rec. (%) | Act. (%) | Rec. (%) |
| 50 cm | 10 | 0 | 100 | 100 | 100 | 100 |
| 30 cm | 45 | 0 | 100 | 100 | 100 | 100 |
| 20 cm | 90 | 0 | 100 | 100 | 100 | 100 |
| 10 cm | 100 | 50 | 100 | 100 | 100 | 100 |
| 5 cm | 100 | 100 | 100 | 100 | 100 | 100 |

**Can We Use Microphones with a Low-gain Amplifier as an Effective Defense?** The preceding discussion may prompt one to propose using microphones with a low-gain amplifier as an effective defense. Unfortunately, this is not true because such microphones require legit users to raise their voices to command the VCS. For example, our experiments show that a user cannot activate Siri from a reasonable distance (2 m) with a soft tone (40 dB) on iPhone 6 Plus. Specifically, we measure the activation rate (i.e., the success rate of activation) and the recognition rate (i.e., the success rate of action commands) of iPhone 6 Plus, iPhone X, and iPhone XR in normal operation environments (i.e., no attacks). We use a Google Pixel phone to replay a normal command “Hey Siri, turn down the volume” to each device at varying distances on the same desk, at a sound pressure level of 40 dB to mimic a human soft tone. Table 6 compares their activation rate and recognition rate, showing that iPhone 6 Plus fails to be controlled by a legit user at a distance of 2 m; whereas, iPhone X and XR can be controlled from a distance of over 5 m. iPhone 6 Plus' poor Siri usability may be the reason why Apple switches to a high-gain amplifier in the later version of iPhones (e.g. 8, X, XR, 13 mini). The experiment video is available on our Demo website [11].

**Insight 5** Siri, Google Assistant, Alexa and Cortana are vulnerable to NUIT attacks, but at different degrees. NUIT (and DolphinAttack with common input) fail to attack iPhone 6 Plus because their microphones use a low-gain amplifier.

## 7 Analyzing the Effectiveness of NUIT

We analyze the impact of the following four factors on the effectiveness of NUIT-1: (i) the action command language, because one action command's lengths are various in different languages (e.g., English vs. French) that may fit into the reactive time window in one case but not another; (ii) the audio file format, because formats impacts sound qualities; (iii) the background noise, because it is often present in practice and should be tolerated (i.e., an attack assuming no background noise is not practical); and (iv) the carrier media audio volume, which may affect the location where the NUIT signals should be embedded. Since the notion of reaction time window doesn't apply to NUIT-2, there is no need to analyze (i) for NUIT-2. This means we only need to analyze the impact

Table 7: Default experimental settings.

| Setting | NUIT-1 | NUIT-2 |
|---|---|---|
| Victim Speaker | iPhone XR | iPhone XR |
| Victim Microphone | | LG ThinkQ |
| Background Noise | 30dB | |
| Activation Keyword | "Hey Siri" | "Hey Google" |
| Action Command | "Turn down the volume" | |
| Distance | N/A | 25cm |
| File format | 16-bit WAV | |
| Carrier Audio | Totally silent | |
| Volume | 80% | |
| Physical Layout | All devices lay on a desk, with screen facing the ceiling | |

of (ii)-(iv) on the effectiveness of NUIT-2. In addition, we consider the following two factors that are unique to NUIT-2: (v) the directionality of the victim microphone to the victim speaker, because it can affect the successful rate when the victim has a different arrangement of device direction; and (vi) the distance between the victim microphone and the victim speaker, which clearly can affect the attack success rate. Table 7 summarizes the experimental settings.

### 7.1 Effectiveness of NUIT-1

#### 7.1.1 Impact of Natural Language

We consider the four most spoken languages [33]: English, Chinese, Spanish, and French. First, we make an audio file of our own activation keyword in each of these languages

Table 8: The voice commands in our experiments, including activation keyword and action commands AC1, AC2, and AC3.

| Natural Language | Act. Keyword | AC1 | AC2 | AC3 |
|---|---|---|---|---|
| English | "Hey Siri" | "Call 1..5..x" | "Turn down the volume" | "Text Sam, I need money" |
| Spanish | "Oye Siri" | "llama al 1..5..x" | "Baja el volumen" | "Envíale un mensaje de texto a Sam, necesito dinero" |
| Chinese | "? Siri" | "\| ë 1..5..x" | " N ó Ï " | "Ù SamÑ í á • ± " |
| French | "Dis Siri" | "Appeler 1..5..x" | "Baisse le volume" | "SMS Sam, j'ai besoin d'argent" |

Table 9: Experimental results show that NUIT-1 succeeds with action commands AC1, AC2, and AC3 in most, but not all, cases of the four languages.

| Natural Languages | AC 1 | AC 2 | AC 3 |
|---|---|---|---|
| English | "Call 1..5" | X | X |
| Spanish | "Call 1..3" | X | X |
| Chinese | "Call 1..9" | X | X |
| French | "Call 1..3" | X | X |

because we are attacking our own device. Second, we prepare Text-To-Speed generated audios of action commands in these languages at 330 words per minute. We consider three examples of action command (AC), which are summarized in Table 8 as AC1, AC2 and AC3, respectively. for AC1, which is “Call + phone number” in English and its equivalent in other languages, we vary the length of the phone number, from 3 to 9 digits because the same command may succeed in some languages but not others.

Table 9 summarizes the experimental results. We observe that for AC1, NUIT-1 successfully calls 9-digit phone numbers in Chinese, 5-digit phone numbers in English, and 3-digit phone numbers in Spanish and French. For AC2, NUIT-1 succeeds in all four languages because the AC2 audios have a similar length (i.e. 0.6 seconds). For AC3, NUIT-1 fails with Spanish commands but succeeds with the other languages. This is because the audio of AC3 in Spanish is 2 seconds, which is longer than reaction time of Siri (0.82 seconds) even at 330 words per minute, but the audio of AC3 in the other languages is at most 0.9 seconds.

Insight 6 The success of NUIT-1 depends on the language because the same action command in different languages can result in audios of different lengths, some of which can t into the reaction time window but others cannot.

#### 7.1.2 Impact of Audio Format

Popular audio formats can be divided into two categories: lossless vs. lossy. A lossless format stores raw audio without any compression, offering the highest audio quality; two examples are Waveform Audio File (wav) and Audio Interchange File Format (AIFF). A lossy format uses compression. Three examples are: MPEG-1 Audio Layer III (mp3), which loses certain components of sound beyond the human hearing frequency range (>16kHz) [34]; Advanced Audio Coding (aac), which has a higher audio quality than mp3 by using a better compression algorithm; and Windows Media Audio (wma), which is similar to mp3. We use the widely-used bitrate of 128 kbps [34] to evaluate mp3, aac, and wma les.

Table 10 summarizes the experimental results. For Siri devices, we observe that attacks leveraging lossless audio les (wav and AIFF) succeed against all listed devices except iPhone 6 Plus. Attacks leveraging lossy audio les (mp3 and wma) always fail Siri devices because these lossy formats cause the elimination of the near-ultrasound attack signals (>16kHz). However, attacks leveraging the lossy aac audio format always succeed against all devices except iPhone 6 Plus. For Google, Alexa, and Cortana devices, we observe that the NUIT-1 attack always succeeds, even if the base audio uses a lossy audio le format. The reason is that Google and Alexa's activation keywords require less bandwidth, which can survive the high frequency loss by mp3 and wma.

Insight 7 For devices vulnerable to NUIT-1 attacks, NUIT-1 attacks succeed when using lossless audio formats, but may fail when using some lossy audio formats.

#### 7.1.3 Impact of Background Noise

To evaluate the impact of background noise on the success of NUIT-1, we use noise to mimic the environment of a bedroom, of ce and cafe. The malicious audio is a 16-bit WAV le. The background noise is some Gaussian White Noise from a Samsung TV speaker in an anechoic chamber at 30dB, 60dB and 70dB, respectively. The noise is generated by a Samsung TV when the victim device replays the malicious audio with embedded NUIT signals. We repeat the attack 100 times to derive the successful rate of the attack (i.e., the percentage of successful attacks over the total number of attacks).

Table 11 summarizes the experimental results. For the NUIT-1 attack, we observe that the background noise mimicking the bedroom (30-45dB) environment or of ce (55-65dB) environment does not have an impact on the attack success rate. However, the noise mimicking cafe environment (65-75dB) causes it to lose effectiveness: 10% of the times the activation keyword fails and 30% of the times the action command fails. The failure can be attributed to the low Signal-to-Noise Rate (SNR), which disrupts the signal even though the speaker and the microphone are on the same device.

Insight 8 The NUIT-1 attack can tolerate certain degrees of background noises because of the short distance between the victim speaker and the victim microphone, but starts to fail when the background noise gets stronger.

Table 10: Effectiveness of NUIT-1, where X means NUIT-1 succeeds, means NUIT-1 fails, N/A means NUIT-1 is not applicable, `AK' means Activation Keyword, `AC' means Action Command, `Volume' is the speaker volume for NUIT-1 to be successful (i.e., the minimum volume at which attacks can succeed).

| Brand | Model | Mobile OS | VCS | AK | AC | Audio Format (kbps) | | | | | Volume |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | wav | mp3 | acc | wma | AIFF | |
| Apple | iPhone XR | iOS 14.8.1 | Siri | X | X | X | | X | | X | 70% |
| | iPhone X | iOS 15.1.1 | Siri | X | X | X | | X | | X | 70% |
| | iPhone 8 | iOS 14.4.2 | Siri | X | X | X | | X | | X | 70% |
| | iPhone 6plus | iOS 13.1.2 | Siri | | | | | | | | 70% |
| | MacBook Pro 2021 | macOS; Monterey | Siri | X | X | X | | X | | X | 75% |
| | MacBook Air 2017 | macOS; Monterey | Siri | N/A | X | X | | X | | X | 75% |
| Samsung | Galaxy S8 | Android 11 | Google | X | X | X | X | X | X | X | 75% |
| | Galaxy S9 | Android 11 | Google | X | X | X | X | X | X | X | 80% |
| | Galaxy A10e | Android 11 | Google | X | X | X | X | X | X | X | 75% |
| Amazon | Echo Dot Gen1 | Fire OS 7 | Alexa | X | X | X | X | X | X | X | 70% |
| Dell | Inspiron 15 | Windows 10 | Cortana | X | X | X | X | X | X | X | 80% |

Table 11: Impact of background noise on the success rate of NUIT-1 and NUIT-2 with the default activation keywords (AK) and action command (AC) described in Table 7.

| Scenario | Noise | Attack Type | AK | AC |
|---|---|---|---|---|
| Bedroom | 30dB | NUIT-1 | 100% | 100% |
| | | NUIT-2 | 100% | 100% |
| Of ce | 60dB | NUIT-1 | 100% | 100% |
| | | NUIT-2 | 100% | 90% |
| Cafe | 70dB | NUIT-1 | 90% | 70% |
| | | NUIT-2 | 80% | 40% |

Table 12: Impact of carrier audio volume on the success of NUIT-1 and NUIT-2 with the default AK and AC.

| Base Volume (dB) | Attack Type | AK | AC |
|---|---|---|---|
| -30 | NUIT-1 | 100% | 100% |
| | NUIT-2 | 100% | 100% |
| 0 | NUIT-1 | 80% | 60% |
| | NUIT-2 | 100% | 100% |
| 30 | NUIT-1 | 20% | 10% |
| | NUIT-2 | 80% | 80% |
| 50 | NUIT-1 | 0% | 0% |
| | NUIT-2 | 40% | 30% |

#### 7.1.4 Impact of Carrier Audio Volume

To evaluate this, we embed NUIT signals into the Gaussian White Noise with sound pressure level 30dB, 0dB, 30dB and 50dB, respectively. This leads to 5 malicious audio les. We repeat each attack 100 times to derive the successful rate. As shown in Table 12 for the NUIT-1 attack when the carrier audio volume is above 0dB, its success rates of the activation keyword and the action command drop from 100% to 80% and 100% to 60%, respectively. This is because the high volume combined with the close proximity between the victim speaker and the victim microphone produces a strong Sound Pressure Level (SPL) at the microphone. This triggers the microphone's Automatic Gain Control (AGC), suppressing the NUIT-1 signal so that Siri cannot understand the command. Moreover, the discrepancy between the 80% and the 60% suggests that even when the activation keyword succeeds, the following action command may fail.

Insight 9 The NUIT-1 attack fails when the attack signals are mixed with carrier audio's sound track.

### 7.2 Effectiveness of the NUIT-2 Attack

#### 7.2.1 Impact of Audio Format

The experimental result is the same as NUIT-1 and thus omitted. This is expected as there is little background noise (<40dB). Thus, we can draw the same insight as Insight 6: audio format signi cantly impacts NUIT-2's success rate.

#### 7.2.2 Impact of Background Noise

To evaluate the impact of background noise, we conduct the same experiments as in the case of NUIT-1, with the difference that we use the NUIT-2 default settings. Experimental results are also summarized in Table 11 for easier comparison. We observe NUIT-2 is more signi cantly affected by the background noise, especially when the noise is loud, because the speaker-microphone distance in the NUIT-1 attack (< 1cm) is much smaller than that of NUIT-2 (25cm).

Insight 10 Background noise has a higher impact on the success of the NUIT-2 attack than the NUIT-1 attack because of the longer speaker-microphone distance in NUIT-2.

#### 7.2.3 Impact of Directionality

Figure 7 shows how we hold the victim speaker device with a phone holder, at coordinate $(x,y,z) = (0,0,0)$. In each experiment, we change the position of the victim microphone device, which is held by hand. Directionality is described by two parameters $\theta$, which is the azimuthal angle [35]; and $\varphi$, which is the polar angle [35]. We vary the $(\theta,\varphi)$ values to observe how they affect the success of NUIT-2. Table 13 summarizes the experimental results. We observe that directionality does not have a signi cant impact on the success rate of NUIT-2: for activation keyword, the attack success rate is always 100%; for action command, the attack success rate is at least 95%. This can be attributed to the omni-directional nature of the near-ultrasound signals.

Figure 7: Illustration of the directionality.

Table 13: Attack success rate of NUIT-2 with varying directionality parameters $(q; j)$ as described in the text. “Cmd" means activation keyword (AK) or action command (AC).

| f \ q | 0 | 45 | 90 | 135 | 180 | Cmd |
|---|---|---|---|---|---|---|
| 0 | 100% | 100% | 100% | 100% | 100% | AK |
| | 95% | 95% | 95% | 95% | 95% | AC |
| 45 | 100% | 100% | 100% | 100% | 100% | AK |
| | 100% | 100% | 95% | 95% | 95% | AC |
| 90 | 100% | 100% | 100% | 100% | 100% | AK |
| | 100% | 100% | 100% | 100% | 95% | AC |
| 135 | 100% | 100% | 100% | 100% | 100% | AK |
| | 100% | 100% | 95% | 95% | 95% | AC |
| 180 | 100% | 100% | 100% | 100% | 100% | AK |
| | 95% | 95% | 95% | 95% | 95% | AC |

Insight 11 Directionality does not have a significant impact on the success of the NUIT-2 attack because of the omnidirectional transmission ability of sound.

### 7.2.4 Impact of Distance

To evaluate the impact of the distance between the victim speaker device and the victim microphone device on the attack success rate of NUIT-2, we vary the distance between them. The experiment setting is the same as the directionality one, except that we vary the distance between the speaker device and the microphone device. We want to determine the effective distance between the speaker device and the microphone device below which the attack success rate is 80%.

Table 14 summarizes the experimental results, which show that the effective distance depends on the power of the speaker. For small mobile devices, the effective distance is small ($<$ 10cm); for devices like laptops, desktops, TVs or car radio, the effective distance can be longer. Moreover, the effective distance of Alexa Echo, Google Home, and Cortana, which do not authenticate activation keywords, is longer than that of Siri and Google Phone Assistant, which authenticate activation keywords. This is because the authentication mechanism does not allow any significant distortion of activation keywords; otherwise, it could be exploited to wage other attacks.

Insight 12 The effective distance between the victim speaker and the victim microphone in the NUIT-2 attack depends on the power of the victim speaker.

## 8 Defense

Security Requirements We propose the following four security requirements for an ideal defense: (i) it detects attacks with few false-positives and few false-negatives; (ii) it is device-independent, meaning the defense can be implemented on any type of modern VCS devices (i.e. mobile, wearable and stationary devices) without modifying/adding existing hardware (iii) it is robust against evasion; (iv) it is light-weight and incurs minimal processing delay. As elaborated in Appendix D, known defenses against previous inaudible attacks cannot be adapted to defeat NUIT. Note that requirement (ii) mandates software solutions.

Our Defense The basic idea is to leverage the success of NUIT attacks to cope with themselves as follows: Whenever the attack succeeds, the victim microphone VCS must have already detected and recognized the embedded NUIT signal at a near-ultrasound frequency; this capability can be leveraged to detect the presence of NUIT because a legitimate activation keyword or action command should not come from the high frequency range ($>$ 16kHz).

Figure 8: Basic idea for detecting NUIT.

Figure 8 highlights the techniques behind the defense. It is based on the following similarity analysis, which is made possible by the nonlinear demodulation, namely that the microphone system produces an inaudible near-ultrasound signal consisting of two parts: the demodulated baseband ($<$ 8kHZ) signal $s_b(t)$ and the high-frequency passband ($>$ 16kHz) signal $s_p(t)$. If $s_b(t)$ comes from $s_p(t)$, then there is a NUIT attack; otherwise, there is no NUIT attack. In greater detail, the defense first divides signal $s_b(t)$ into segments of fixed-length $T_{win}$. The windowed commands are filtered by a Low-Pass Filter (LPF) with a cut-off frequency 16kHz and a High-Pass Filter (HPF) also with a cut-off frequency 16kHz. The signal passing the HPF has a high frequency ($>$ 16kHz) which will be squared and compared with the baseband signal using cross-correlation with coefficient $R$ with

$$R = \frac{1}{T}\int_{t_0}^{t_0+T_{win}} s_b(t)\, s_p^2(t)\, dt.$$

A similarity threshold $t$ can be used such that $|R| > t$ means that a NUIT attack is detected. This is because a high similarity between the envelope of the high frequency component ($>$

Table 14: Effectiveness of NUIT-2, where each cell describes the maximum distance (in centimeters) between the victim speaker device and the victim microphone device at which NUIT-2 succeeds with effectiveness 80%, and means NUIT-2 fails.

| Victim Speaker \ Victim Microphone | | Siri: iPhone XR | Siri: MacBook Pro-2021 | Siri: Apple Watch 2 | Google Phone Assistant: Google Pixel 3 | Google Phone Assistant: Galaxy S9 | Google Phone Assistant: LG Think Q V35 | Google Phone Assistant: Galaxy Tab S4 | Alexa: Echo Dot Gen 1 | Google Assistant: Google Home 2 | Cortana: Dell Inspiron 15 | Cortana: MS Surface |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Apple Devices | iPhone XR | 3 | 3 | 3 | 4 | 6 | 50 | 5 | 6 | 7 | 6 | 8 |
| | MacBook Pro | 9 | 8 | 10 | 20 | 25 | 130 | 20 | 30 | 25 | 310 | 320 |
| | iPhone13 mini | 3 | 3 | 3 | 4 | 6 | 50 | 5 | 5 | 7 | 6 | 8 |
| | iMac 27' 2021 | 13 | 12 | 15 | 13 | 30 | 390 | 20 | 50 | 60 | 370 | 350 |
| Android Devices | LG Think Q V35 | | | | | | | | | | | |
| | Samsung Galaxy S9 | 4 | 4 | 4 | 6 | 4 | 60 | 6 | 7 | 5 | 7 | 7 |
| | Samsung Galaxy Tab S4 | 9 | 9 | 10 | 27 | 20 | 150 | 20 | 40 | 50 | 25 | 30 |
| Vehicle Audio Sys. | Ford Fusion 2017 | 30 | 28 | 35 | 102 | 82 | 320 | 70 | 210 | 230 | 160 | 140 |
| | Nissan Versa S | | | | 110 | 70 | 300 | 65 | 190 | 220 | 150 | 150 |
| Smart Home Devices | Samsung TV | 35 | 32 | 46 | 120 | 80 | 460 | 90 | 350 | 320 | 150 | 100 |
| | Google Home2 | 3 | 2 | 2 | 15 | 25 | 380 | 27 | 38 | 39 | 58 | 60 |
| | Echo Dot Gen1 | 2 | 1 | 1 | 17 | 29 | 320 | 26 | 42 | 33 | 62 | 69 |
| Windows Laptop | Dell Inspiron15 | | | | 25 | 20 | 300 | 25 | 90 | 100 | 50 | 45 |

16kHz) and the waveform of the baseband component (< 16kHz) will make the command shadowed from the high frequency range, indicating the presence of attacks.

**Defense Effectiveness Analysis.** Since the defense is software-based, it can be implemented on any existing device without modifying or adding any hardware, satisfying requirement (ii). Since the attacker cannot decrease the similarity, the defense is robust against evasion or satisfies requirement (iii). The other security requirements are satisfied as evidenced by the following experiment-based evaluation.

We record 300 instances of activation keywords for iPhone XR, including 100 from a human at a distance of 5cm, 100 NUIT-1 signals from its speaker, and 100 NUIT-2 signals from a Samsung S9 at a distance of 5cm. (These devices are arbitrarily chosen because all microphones follow the same nonlinearity principle.) For speech processing, $T_{win}$ is 20ms-40ms [36]; we choose 40ms to better capture low-frequency characteristics [36]. We set $t = 0:55$. The 200 malicious audios and the 100 legit command audios are waged against our defense in the setting mentioned in Section 7. Figure 9 summarizes the experimental results, showing the defense achieves zero false-positives and zero false-negatives, satisfying requirement (i). The defense is a light-weight, satisfying requirement (iv). In summary, the defense satisfies all of the four requirements mentioned above.

Figure 9: The defense achieves zero false-positives and zero false-negatives in 300 experiments, where $t = 0:55$.

## 9 Limitation

The study has several limitations. (i) The inaudibility of NUIT attacks is rooted in the inaudibility of near-ultrasound signals. However, some young people may be able to hear near-ultrasound sound, meaning that NUIT may be audible to them. Nevertheless, NUIT can attack most users. (ii) The success rate of NUIT would be affected by the quality of the victim speakers as evidenced by our experiment that the LG Think Q V35 speaker has a poor response above 16kHz and thus cannot be exploited to wage the NUIT attack. (iii) For NUIT to succeed, the victim speaker must be above a certain volume level; otherwise, the attack will fail. (iv) The NUIT-1 end-to-end unnoticeability (i.e., inaudible attack and silent device response) is not universally true but depends on how the device response mechanism is implemented. (v) The NUIT-1 attack is inherently limited by the reaction time (< 1s), making it impossible to inject long action commands that cannot be split into multiple short commands. (vi) The NUIT-1 attack fails to attack devices with a low-gain microphone (i.e., iPhone 6 Plus). (vii) The NUIT-2 attack requires a short distance between the victim speaker and the victim microphone, especially for low-power speaker devices (e.g., smartphones.) (viii) The NUIT-2 attack may fail when the victim's speaker device has the same VCS as the targeted microphone device, because it may trigger NUIT-1 attack on the speaker device.

## 10 Conclusion

We have introduced NUIT, which is a new class of inaudible attacks against VCSs and can be waged remotely. Unlike previous inaudible attacks, NUIT exploits victim speakers to attack victim microphones and associated VCSs. To realize NUIT, we address three challenges and our ideas may be of independent value. We demonstrate the feasibility of NUIT and propose a novel and effective defense against NUIT. We hope this study will inspire more research on VCS security, for which the limitations of this study can be a starting point.

**Acknowledgments.** We thank the anonymous reviewers for their comments that guided us in revising the paper. This work was supported in part by the U.S. Department of Energy/National Nuclear Security Administration (DOE/NNSA) #DE-NA0003985, NSF Grants #2122631 and #2115134, and Colorado State Bill 18-086. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any of these funding agencies.

## References

[1] N. Roy, H. Hassanieh, and R. Roy Choudhury, "Backdoor: Making microphones hear inaudible sounds," in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 2–14, 2017.

[2] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "Dolphinattack: Inaudible voice commands," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 103–117, 2017.

[3] L. Song and P. Mittal, "Poster: Inaudible voice commands," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2583–2585, 2017.

[4] N. Roy, S. Shen, H. Hassanieh, and R. R. Choudhury, "Inaudible voice commands: The long-range attack and defense," in *15th USENIX Symposium on Networked Systems Design and Implementation*, pp. 547–560, 2018.

[5] T. Takeshi, C. Benjamin, R. Sara, et al., "Light commands: laser-based audio injection attacks on voice-controllable systems," 2019.

[6] X. Yuan, Y. Chen, Y. Zhao, Y. Long, X. Liu, K. Chen, S. Zhang, H. Huang, X. Wang, and C. A. Gunter, "Commandersong: A systematic approach for practical adversarial voice recognition," in *27th USENIX Security Symposium (USENIX Security 18)*, pp. 49–64, 2018.

[7] T. Chen, L. Shangguan, Z. Li, and K. Jamieson, "Metamorph: Injecting inaudible commands into over-the-air voice controlled systems," in *27th Annual Network and Distributed System Security Symposium, NDSS 2020, San Diego, California, USA, February 23-26, 2020*, The Internet Society, 2020.

[8] R. E. Ziemer and W. H. Tranter, *Principles of communications*. John Wiley & Sons, 2014.

[9] "Tukey window." https://en.wikipedia.org/wiki/Window_function. Accessed: 2023-1-30.

[10] Q. Yan, K. Liu, Q. Zhou, H. Guo, and N. Zhang, "Surfingattack: Interactive hidden attack on voice assistants using ultrasonic guided waves," in *Network and Distributed Systems Security (NDSS) Symposium*, 2020.

[11] "Nuit demo weblink." https://sites.google.com/view/nuitattack/home. Accessed: 2023-1-30.

[12] X. Ji, J. Zhang, S. Jiang, J. Li, and W. Xu, "Capspeaker: Injecting voices to microphones via capacitors," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1915–1929, 2021.

[13] S. Ka, T. H. Kim, J. Y. Ha, S. H. Lim, S. C. Shin, J. W. Choi, C. Kwak, and S. Choi, "Near-ultrasound communication for tv's 2nd screen services," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pp. 42–54, 2016.

[14] N. Karapanos, C. Marforio, C. Soriente, and S. Capkun, "Sound-Proof: Usable Two-Factor authentication based on ambient sound," in *24th USENIX Security Symposium (USENIX Security 15)*, pp. 483–498, 2015.

[15] G. E. Santagati and T. Melodia, "U-Wear: Software-defined ultrasonic networking for wearable devices," in *Proceedings of the 13th annual international conference on mobile systems, applications, and services*, pp. 241–256, 2015.

[16] H. Lee, T. H. Kim, J. W. Choi, and S. Choi, "Chirp signal-based aerial acoustic communication for smart devices," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 2407–2415, IEEE, 2015.

[17] Y. Bai, J. Liu, L. Lu, Y. Yang, Y. Chen, and J. Yu, "Batcomm: enabling inaudible acoustic communication with high-throughput for mobile devices," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, pp. 205–217, 2020.

[18] C. Kasmi and J. L. Esteves, "IEMI threats for information security: Remote command injection on modern smartphones," *IEEE Transactions on Electromagnetic Compatibility*, vol. 57, no. 6, pp. 1752–1755, 2015.

[19] C. Kasmi and J. L. Esteves, "Whisper in the wire: Voice command injection reloaded," Hack In Paris, 2016.

[20] T. Vaidya, Y. Zhang, M. Sherr, and C. Shields, "Cocaine noodles: Exploiting the gap between human and machine speech recognition," in 9th USENIX Workshop on Offensive Technologies, WOOT '15, Washington, DC, USA, August 10-11, 2015 (A. Francillon and T. Ptacek, eds.), USENIX Association, 2015.

[21] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. A. Wagner, and W. Zhou, "Hidden voice commands," in 25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016 (T. Holz and S. Savage, eds.), pp. 513–530, USENIX Association, 2016.

[22] Y. He, J. Bian, X. Tong, Z. Qian, W. Zhu, X. Tian, and X. Wang, "Canceling inaudible voice commands against voice control systems," in The 25th Annual International Conference on Mobile Computing and Networking, pp. 1–15, 2019.

[23] C. Wang, S. A. Anand, J. Liu, P. Walker, Y. Chen, and N. Saxena, "Defeating hidden audio channel attacks on voice assistants via audio-induced surface vibrations," in Proceedings of the 35th Annual Computer Security Applications Conference, pp. 42–56, 2019.

[24] Y. Meng, J. Li, M. Pillari, A. Deopujari, L. Brennan, H. Shamsie, H. Zhu, and Y. Tian, "Your microphone array retains your identity: A robust voice liveness detection system for smart speakers," in 31st USENIX Security Symposium (USENIX Security 22), (Boston, MA), USENIX Association, Aug. 2022.

[25] G. Zhang, X. Ji, X. Li, G. Qu, and W. Xu, "Eararray: Defending against dolphinattack via acoustic attenuation," in Network and Distributed Systems Security (NDSS) Symposium, 2021.

[26] Y. Lee, Y. Zhao, J. Zeng, K. Lee, N. Zhang, F. H. Shezan, Y. Tian, K. Chen, and X. Wang, "Using sonar for liveness detection to protect smart speakers against remote attackers," Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., vol. 4, no. 1, pp. 16:1–16:28, 2020.

[27] S. A. Gelfand, Essentials of Audiology. Thieme, 2011.

[28] J. Yang, S. Sidhom, G. Chandrasekaran, T. Vu, H. Liu, N. Cecan, Y. Chen, M. Gruteser, and R. P. Martin, "Detecting driver phone use leveraging car speakers," in Proceedings of the 17th annual international conference on Mobile computing and networking, pp. 97–108, 2011.

[29] R. V. Cox, S. F. D. C. Neto, C. Lamblin, and M. H. Sherif, "Itu-t coders for wideband, superwideband, and fullband speech communication [series editorial]," IEEE Communications Magazine, vol. 47, no. 10, pp. 106–109, 2009.

[30] H. Landau, "Sampling, data transmission, and the nyquist rate," Proceedings of the IEEE, vol. 55, no. 10, pp. 1701–1706, 1967.

[31] "The best voice assistant." http://dx.doi.org/10.1002/andp.19053221004. Accessed: 2023-1-30.

[32] "Avisoft vifa speaker." http://www.avisoft.com/. Accessed: 2023-1-30.

[33] "Most spoken language." https://www.berlitz.com/en-uy/blog/most-spoken-languages-world, 2021. Accessed: 2023-1-30.

[34] "mp3 format." https://docs.fileformat.com/audio/mp3/, 2021. Accessed: 2023-1-30.

[35] "Weisstein, eric w. "spherical coordinates." from mathworld–a wolfram web resource.." https://mathworld.wolfram.com/SphericalCoordinates.html, 2023. Accessed: 2023-1-30.

[36] K. K. Paliwal, J. G. Lyons, and K. K. Wójcicki, "Preference for 20-40 ms window duration in speech analysis," in 2010 4th International Conference on Signal Processing and Communication Systems, pp. 1–4, IEEE, 2010.

[37] J. Liu, Nonlinear dynamics of a dual-backplate capacitive MEMS microphone. PhD thesis, Citeseer, 2007.

[38] S. A. Zawawi, A. A. Hamzah, B. Y. Majlis, and F. Mohd-Yasin, "A review of mems capacitive microphones," Micromachines, vol. 11, no. 5, p. 484, 2020.

[39] I. Otung, Communication engineering principles. John Wiley & Sons, 2021.

[40] L. R. Rabiner and B. Gold, "Theory and application of digital signal processing," Englewood Cliffs: Prentice-Hall, 1975.

[41] M. E. Ahmed, I.-Y. Kwak, J. H. Huh, I. Kim, T. Oh, and H. Kim, "Void: A fast and light voice liveness detection system," in 29th USENIX Security Symposium (USENIX Security 20), pp. 2685–2702, 2020.

# Appendix A COTS Speaker Frequency Response

Figure 10 plots the experimental results of the frequency response of Samsung Galaxy S10, iPhone 7, and Google Pixel

3 speakers, in terms of normalized sound pressure (with the maximum amplitude set to 0dB). We observe that different speakers have different high frequency responses. In particular, speakers can send near-ultrasound high frequency signals (16kHz-22kHz) with some deterioration when compared with the audible frequency range (20Hz-16kHz), meaning that `NUIT` can exploit the 6kHz (i.e., 16kHz-22kHz) to wage inaudible attacks.

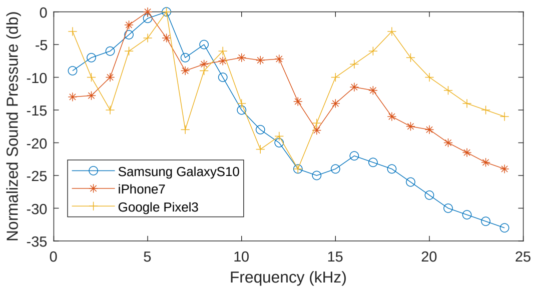



Figure 10: Empirical frequency response of COTS speakers.

## Appendix B Why Isn't DSB-AM Applicable to NUIT?

In order to explain why the inaudible *airborne* ultrasound attacks [1, 2, 4] are *not* applicable to the setting of `NUIT`, we first review how these attacks operate. They proceed in three steps. (i) The attacker uses the DSB-AM scheme to modulate audible voice commands (at a frequency <16kHz) to an inaudible ultrasound frequency (i.e.,  20kHz). The modulated signals contain two sidebands with a total passband bandwidth of 16kHz (i.e., one sideband needs 8kHz to attack VCS devices). (ii) The attacker emits inaudible ultrasound signals by using one or multiple (possibly an array of) ultrasonic transducers, which are owned and operated by the attacker, to the victim device's microphone. (iii) After the victim device's microphone receives the ultrasound signal, the microphone automatically demodulates the ultrasound signal back to voice command signals to activate the VCS. This is made possible by a physical property of microphones, known as *nonlinearity*, which is an inherent physical property that has been exploited by previous inaudible attacks and is also exploited by `NUIT`. Details follow.

Modern VCS uses Micro-ElectroMechanical System (MEMS) microphones to convert acoustic vibrations or sound waves to electrical signals. When an incoming acoustic signal, denoted by $s_{in}$, is received by the membrane and capacitor, it is transformed into a weak electrical signal, which is then amplified by a pre-amplifier module and fed into a Low-Pass Filter (LPF). The LPF removes inaudible noises with frequency > 20kHz and then sends the audible signal to an Analog-to-Digital Converter (ADC). The ADC outputs a quantized output signal, denoted by $s_{out}$, which is to be processed by VCS. Let $A_1$ and $A_2$ respectively denote the coefficients of the linear term and the nonlinear terms. When the input signal is amplified, the nonlinearity of the microphone cannot be ignored [37, 38]. By omitting the higher-order terms whose coefficients are close to 0 [37, 38], the output signal becomes

$$s_{out}(t) \quad A_1 s_{in}(t) + A_2 s_{in}^2(t);$$

where the term $s_{in}^2(t)$ contributes to the nonlinear demodulation of the input signals that were modulated by DSB-AM. Let $v(t)$ denote the baseband signal (i.e., voice commands). The DSB-AM modulated signal corresponding to an inaudible command sent by the ultrasonic transducer is expressed as

$$s_{in}(t) = (1 + v(t))cos(2p f_c t);$$

where $f_c$ denotes the ultrasonic carrier frequency (i.e., $f_c >$ 20kHz). After the microphone's processing, the signal contained in $f_c$ is filtered as mentioned above, meaning that the demodulated signal received by the VCS is

$$s_{out}(t) = A_2(1 + 2v(t) + v(t)^2)=2; \quad (4)$$

where the $v(t)$ component contributes to VCS' recognition of $s_{out}$ as a legitimate voice command.

In summary, by taking advantage of a victim microphone's nonlinearity property, DSB-AM can be used to attack VCS devices with a passband bandwidth of 16kHz.

## Appendix C Eliminate Burst Noise

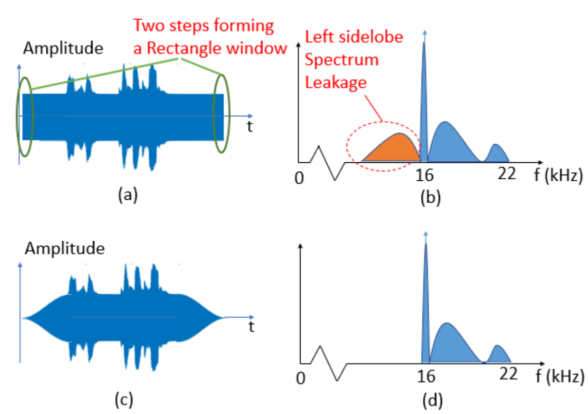



Figure 11: The cause and elimination of burst noises: (a) Raw $S_{USBAM}(t)win_{base}(t)$ in time domain; (b) Frequency spectrum of $S_{USBAM}(t)win_{base}(t)$; (c) $S_{USBAM}(t)\mathrm{T}\kappa(t)$ in time domain; (d) Frequency spectrum of $S_{USBAM}(t)\mathrm{T}\kappa(t)$.

**Root Cause of Burst Noises**. Raw `NUIT` signals may incur burst noises if replayed on COTS speakers without smoothing steps. This phenomenon is known as *spectral leakage* [39, pp. 285]. A raw SSB-AM signal has two sharp steps at its two ends, as illustrated in Figure 11. These steps form a time-domain rectangle window $win_{base}$. A USB-AM signal with these steps can be expressed as:

$$\begin{aligned} & S_{USBAM}(t)win_{base}(t) \quad (5) \\ = & [(1 + v(t))\cos(2\pi f_c^u t) - \hat{v}(t)\sin(2\pi f_c^u t)]win_{base}(t), \end{aligned}$$

where $win_{base}$ is a rectangle window of length $L$ and

$$win_{base} = \begin{cases} 1 & 0 \le t \le L \\ 0 & otherwise. \end{cases}$$

Since the frequency spectrum of $win_{base}$ is a sample function $sinc(f)$ [8, pp. 30], the component $win_{base}cos(2\pi f_c^u t)$ in Eq.(5) has a spectrum of a sampling function with the center frequency raised to $f_c^u$, namely $sinc(f - f_c^u)$. Since $f_c^u = 16kHz$ in this paper, the left-side lobe of $sinc(f - f_c^u)$ goes into the audible frequency range ($< 16$kHz), causing audible burst noises.

**Eliminating Burst Noises Caused by Spectral Leakage**. Having pinned down the root cause of burst noises, we propose eliminating them by suppressing the side lobe without deforming the NUIT signal. For this purpose, we multiply the modulated signal by a *Tukey window* $\mathrm{T}\textsc{k}$, which is also known as the cosine-tapered window [40], before embedding a NUIT signal into a carrier audio $S_{USBAM}(t)\mathrm{T}\textsc{k}(t)$. Recall that

$$\mathrm{T}\textsc{k} = \begin{cases} \frac{1}{2}(1 + cos(\frac{2\pi}{\alpha}(t - \frac{\alpha}{2}))) & 0 \le t < \alpha/2 \\ 1 & \alpha/2 \le t \le 1 - \alpha/2 \\ \frac{1}{2}(1 + cos(\frac{2\pi}{\alpha}(t - 1 + \frac{\alpha}{2}))) & t > 1 - \alpha/2 \end{cases}$$

for some $0 < \alpha < 1$ [40]. A larger $\alpha$ reduces more spectral leakage, but requires a slower rolling-down (i.e., a longer unmodulated part of the signal at each end). This means that the attacker needs to make a trade-off between the length of the unmodulated part of the signal and the spectral leakage: an SSB-AM signal with long unmodulated parts at either end may waste valuable time for injecting NUIT signals, but long unmodulated parts make the Tukey window roll down more slowly, reducing spectrum leakage. Our experiments show:

**Insight 13** *Multiplying the raw* NUIT *signal with Tukey Window and setting its* $\alpha > 0.5$ *can eliminate burst noises.*

# Appendix D Why Are Known Defenses Ineffective against NUIT?

This section elaborates on why known defenses cannot defeat NUIT. We divide known defenses into two categories: Multi-factor defenses vs. Single-factor defenses.

## D.1 Why Are Known Multi-factor Defenses Ineffective against NUIT?

At a high level, these defenses rely on the victim device's other hardware than the microphone (e.g. motion sensors [23], microphone array [24, 25], extra speakers [22]) to pick up the voice commands' features in the relevant domain (e.g. vibration spectrum [23], directionality [25], acoustic field distribution [24], or user's physical location [26]). These defenses have the limitation that the victim VCS device must contain such additional hardware, and are not applicable to devices without such hardware, violating Security Requirement (ii) specified in Section 8. That is, these defenses are ineffective against NUIT attacks.

Specifically, Surface Vibration [23] extracts audio-induced surface vibration features as an additional factor to defend against audible/inaudible attacks. However, this defense relies on motion sensors (e.g. accelerators, gyroscopes) to pick up the surface vibration features, making this defense only applicable to mobile devices and wearable devices, but not stationary VCS devices without motion sensors (e.g. Google Home, Alexa Echo). [24, 25] both use a microphone array to capture the sound field and the acoustic attenuation rate to detect attacks. However, these defenses rely on a microphone array, which is not applicable to most mobile/wearable devices that contain only one microphone (e.g., smart phone, smart watch). [26] leverages network-connected speakers to build a sonar-like system to detect the user's AoA (angle of arrival) for liveness detection. However, this sonar-like system requires extra speakers.

## D.2 Why Are Known Single-factor Defenses Ineffective against NUIT?

We further divide single-factor defenses into two subcategories: hardware-based vs. software based.

**Limitations of Hardware-based Single-factor Defenses.** [22] uses extra ultrasonic transducers to generate a guard signal to actively cancel out the inaudible ultrasonic attack signal. However, the guard signal generator is extra hardware that is not equipped with most modern VCS devices. This violates Security Requirements (ii) specified in Section 8. That is, these defenses are ineffective against NUIT attacks.

**Limitations of Software-based Single-factor Defenses.** Existing software-based single-factor defenses detect "abnormal" behavior in the frequency domain of audio received by a microphone to detect attack signals. These defenses satisfy the following three Security Requirements specified in Section 8: (i), meaning few false-positives and few false-negative; (ii); meaning achieving device-independence, and (iv); meaning lightweight. However, these defenses can be evaded by a crafty attacker, violating Security Requirement (iii). That is, these defenses are ineffective against NUIT attacks. Details follow.

The first approach to software-based single-factor defense leverages speaker characteristics via the spectrum of single-channel audio to detect the liveness of a command and thus attack signals [41]. However, this approach fails to detect attacks waged from good quality speakers with flat frequency