EchoSafe: Sonar-based Verifiable Interaction with Intelligent Digital Agents

Amr Alanwar
University of California, Los Angeles
Los Angeles, California, USA
alanwar@ucla.edu

Bharathan Balaji
University of California, Los Angeles
Los Angeles, California, USA
bbalaji@ucla.edu

Yuan Tian
Carnegie Mellon University
Moffett Field, USA
yt@cmu.edu

Shuo Yang
University of California, Los Angeles
Los Angeles, California, USA
sunnieryounger@ucla.edu

Mani Srivastava
University of California, Los Angeles
Los Angeles, California, USA
mbs@ucla.edu

ABSTRACT

Voice controlled interactive smart speakers, such as Google Home, Amazon Echo, and Apple HomePod are becoming commonplace in today's homes. These devices listen continually for the user commands, that are triggered by special keywords, such as "Alexa" and "Hey Siri". Recent research has shown that these devices are vulnerable to attacks through malicious voice commands from nearby devices. The commands can be sent easily during unoccupied periods, so that the user may be unaware of such attacks. We present *EchoSafe*, a user-friendly sonar-based defense against these attacks. When the user sends a critical command to the smart speaker, EchoSafe sends an audio pulse followed by post processing to determine if the user is present in the room. We can detect the user's presence during critical commands with 93.13% accuracy, and our solution can be extended to defend against other attack scenarios, as well.

CCS CONCEPTS

• **Computer systems organization** → **Embedded software**; • **Security and privacy** → *Multi-factor authentication*;

KEYWORDS

Security, smart home, internet of things, alexa, echo, siri, smart speakers

ACM Reference Format:
Amr Alanwar, Bharathan Balaji, Yuan Tian, Shuo Yang, and Mani Srivastava. 2017. EchoSafe: Sonar-based Verifiable Interaction with Intelligent Digital Agents. In *Proceedings of Proceedings of the First ACM Workshop on the Internet of Safe Things (SafeThings'17).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3137003.3137014

SafeThings'17, November 5, 2017, Delft, The Netherlands

ACM ISBN 978-1-4503-5545-2/17/11…$15.00
https://doi.org/10.1145/3137003.3137014

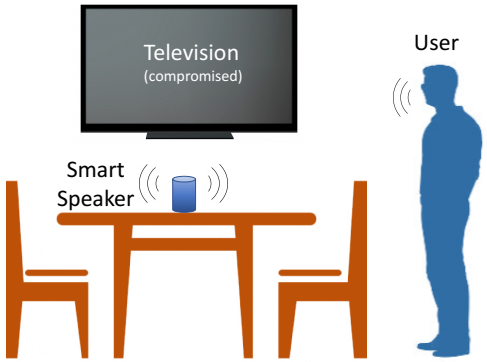

Figure 1: EchoSafe Overview

1 INTRODUCTION

Voice interaction based smart speakers, such as Amazon Echo, have emerged, as a popular way to interact with smart home devices in a hands free manner. More than 11 million Echo devices were sold as of 2017 [24]. Also, major vendors such as Google, Apple, Nvidia, and Bosch have introduced there own devices.

Smart speakers consist of an omnidirectional speaker and a microphone array to listen to the user from any direction. The speaker is in listening mode all the times. It only activates the device to listen for commands, when a preselected keyword such as Alexa or Siri is used. Once the speaker is activated, the microphone array uses beamforming to increase audio sensitivity in the direction of the speaker. The received voice command is sent to a cloud service, where it is transcribed using speech recognition algorithms. The command is parsed, executed, and the results are sent back to the user as an audio response. The speaker connects to Internet via home WiFi and uses user accounts for authentication. Smart speakers provide a variety of services, such as providing general information, setting reminders (e.g. for medicine), controlling smart home devices, placing online orders, and integrating third party applications.

As the smart speakers are always listening, they are susceptible to security attacks by devices that can generate malicious voices. Audio from television news triggered Amazon Echo to place orders for dollhouse [25]. Recent research has shown that machine learning models can mis-classify an input, if an adversary carefully

adds noise to its input [7]. Also, Goodfellow et al. in the same work show that an image of panda is misclassified as a gibbon with high confidence after adding noise by an adversary, but to a human the image still looks like a panda. Carlini et al. [3] build on this idea, and create malicious commands for speech recognition algorithms commonly used in smart speakers and smartphones. Humans hear the audio commands generated as garbled sounds while the speech recognition algorithms interpret them as commands. The authors posit that such commands can be embedded into online videos or TV advertisements to attack devices.

Roy et al. [21] introduce a different type of attack. They identify that the microphones used in modern devices have non-linearities associated with translating analog sounds to digital audio. These non-linearities are benign and filtered away when the microphone receives sounds in the expected frequency range of 20Hz to 20kHz. However, if the input to the microphone is ultrasound (i.e. frequency >20kHz), the microphone can interpret the sound in the audible range. The authors exploit these non-linearities to create ultrasound signals that are inaudible to humans but interpreted as normal sounds by the device. They use this exploit to block the microphone from working (e.g. in a movie theater), and transfer data between devices. We posit that the same methods can be used to send malicious commands to smart speakers without the users being aware of them.

Several types of defenses have been proposed. Amazon Echo has an option to add a pin for making purchases. Carlini et al. [3] propose multiple defense options – the smart speaker can provide audio/visual feedback on reception of a command, an audio captcha can be used to verify if the command is from the user, a machine learning classifier can distinguish between human and machine generated commands. However, these defenses are not fool proof, they add a burden on the usability of the device. Also, the authors themselves conclude that further research is needed. We provide a detailed comparison of different defenses in the Related Work section.

We propose an active defense mechanism called **EchoSafe**, where the smart speaker uses *sonar* (SOund Navigation And Ranging[1]) to verify the availability of the user during critical commands. When the smart speaker receives a command that requires authentication, it sends out an omnidirectional audio signal and analyzes the reflections received by the microphone array to sense the person. Sonar is routinely used in underwater applications, such as submarine navigation. Furthermore, recent research has used the same techniques to map the shape of a room [6], and recognize gestures using smartphones [16].

In this paper, we present a proof of concept of EchoSafe by demonstrating that sonar can be used to verify the presence of a person in a room. We can build upon this idea to mount stronger defenses, such as using a gesture based password or even a gesture captcha to deter sophisticated attacks. Such methods would provide security equivalent to the two factor authentication, while still maintaining the convenience of hands free interaction. Sonar can also be used for other applications that rely on the presence detection such as occupancy based control [14, 22], and the fine-grained signal processing techniques can be exploited for activity recognition, as well[2].

In our proof of concept experiments, we setup a prototype smart speaker to send audio signals both in presence and absence of a person, and exploit machine learning classification with relevant features to detect user presence. We find that EchoSafe can detect user presence with an accuracy of 93.13%.

2 THREAT MODEL

We focus on remote attacks through compromised devices like television, speakers, or smartphones that can generate malicious voice commands in the absence of the user. We trust the smart speaker device manufacturer and third party service providers. The attacks on cloud services and compromised communication links are out of the scope of this work. We consider attacks in a single room scenario and assume that there can only be a single user in the line of sight of the smart speaker. We assume attacks in presence of the user will be easily detected through audio/visual feedback. The user and the attacker can give commands from any direction, smaller objects such as papers, chairs can be moved around, and there can be external ambient sounds.

3 METHODOLOGY

EchoSafe aims to detect the availability of the user in the room, when a critical command is received by the smart speaker in order to protect against many powerful attacks. The intuition behind EchoSafe is to generate a sound pulse from the smart speaker, then capture the echoes reflected from the objects and humans in the room. We use machine learning classifiers to detect the difference between the presence and absence of the user based on features selected from the reflected audio signals received after the smart speaker emits an omnidirectional sound. There is a training phase where the user provides ground truth labels of occupancy in the room, and the smart speaker trains its classifiers based on multiple rounds of emitting sounds and analyzing the received audio. After the training phase, the speaker emits a sound every time it receives a critical voice command, sends the received audio signals to the classifier, and executes the command only when it verifies the availability of a person successfully.

EchoSafe uses Random Forest machine learning algorithm for detecting the person availability. We divide the audio time series data from each microphone into different time windows, then use standard deviation of each time window, and the mel-frequency cepstrum coefficients (MFCC), as candidate machine learning classification features. Picking the most relevant features among the candidates is needed to avoid the curse of dimensionality, speed up the learning process, and achieve better machine learning models. Therefore, the Relief-F algorithm [13] is used to perform the feature selection process among the candidates features efficiently.

4 EXPERIMENT SETUP

We created our own prototype of a smart speaker in order to conduct experiments. Our prototype consists of Matrix Creator[3] which

[1] https://en.wikipedia.org/wiki/Sonar

[2] https://en.wikipedia.org/wiki/Activity_recognition

[3] Matrix Creator: https://creator.matrix.one/

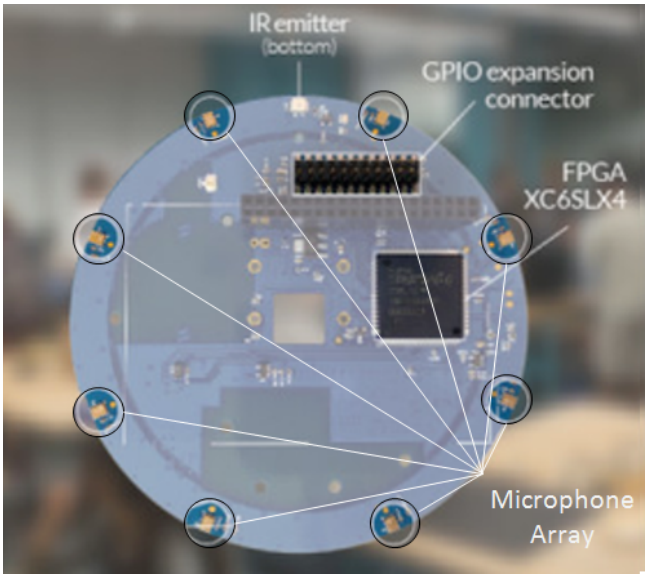

Figure 2: The Matrix Creator Board. It consists of 8 microphones, 32 multi-color LEDs, an FPGA and a GPIO connector that is attached to Raspberry Pi 3 in our experimental setup.

Figure 3: The experiment portion of our lab. The area measures 6.2m x 3.6m and is part of the larger lab measuring 9m x 10m. During the experiments chairs were moved around, and there was ambient noise from activities in the rest of the lab.

is a microphone array available in the market, a small omnidirectional speaker and a Raspberry Pi 3. Figure 2 shows the Matrix Creator with 8 microphones in a circular array, an FPGA for audio processing and a GPIO header that connects to Raspberry Pi in our setup. Sampling the microphones signal is done at the standard 40kHz frequency, so that, it can capture all the frequencies in the audio spectrum (i.e. 20Hz to 20kHz). The microphone array together performs beamforming and produces an additional audio signal apart from the 8 signals received from the individual microphones. We use all the nine audio signals in our experiments.

We use a rectangular portion of our research lab for experiments, two of the sides are regular walls while the other two sides have short cubicle divisions. The dimensions of the experiment area is 6.2x3.6m^2 which is almost an open area as a part of 9x10m^2 lab as shown in Figure 3. There is a rectangular table with chairs around it in the middle of the area, and we place our EchoSafe prototype at the center of the table. The user stands somewhere within this experiment area when they are supposed to be present in our experiments. Other furnitures include a sofa, a television and a bookshelf. We present the details of the experiments and results analysis in the Evaluation section.

5 EVALUATION

We send a 1kHz tone for about one second using the omnidirectional speaker, then acquire the audio signals from Matrix Creator for four seconds. Raspberry Pi is used to coordinate, keep time and store the results. We refer to this brief experiment as a *sonar cycle*. We collect data from several sonar cycles both in presence and absence of the user. We use the data collected to train and test our machine learning models. We tried several features: maximum, mean, standard deviation, percentiles, skewness, kurtosis and MFCC. MFCC is a popular feature for audio signals and captures characteristics of the frequency spectrum. Skewness captures the symmetry of the probability distribution of the signal and kurtosis captures the long tail characteristics of the probability distribution. The rest of the features are regular timeseries characteristics and are self-explanatory. Note that we record 9 audio signals for each sonar cycle, so each of these signals has their corresponding feature set. In addition, we divide each audio signal into smaller time windows to capture the time varying characteristics. We implement both feature extraction and machine learning models using Matlab.

In order to evaluate EchoSafe, we perform a series of experiments. For our initial machine learning classification, we collected 400 sonar cycle data. During 200 sonar cycles no one was there in the experiment area. On the other hand, 200 sonar cycles were collected while some one stood in different positions of the experimental area. We conducted the experiments over different days and the furniture in the testing area were moved randomly. Some students were there in the remaining part of lab conducting their work and talking randomly during all the experiments. The presence of the working students in the remaining part of the lab can be presented as a background noise in real life scenarios.

We evaluated the performance of different supervised learning classifiers to detect the availability of someone in the experimental area. Each microphone output is divided into 8 time windows. We select the most relevant 50 features using the Relief-F algorithm [13]. We tried using the random forest classifier (RF) and support vector machines classifier (SVM) with quadratic kernel, order three polynomial kernel, gaussian radial basis function kernel (RBF), and

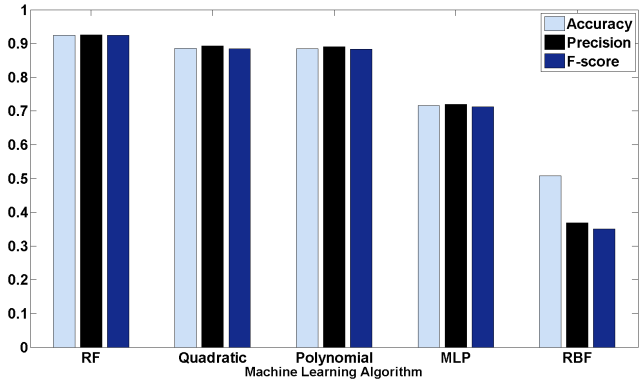

Figure 4: Comparison of cross-validation performance of machine learning classifiers in EchoSafe. We used 400 sonar cycle data for the experiment, where each sonar cycle consists of 1 second of speaker sound and 4 seconds of microphones listening. We divided the 4 second data into 8 time windows and chose 50 statistical features using Relief-F.

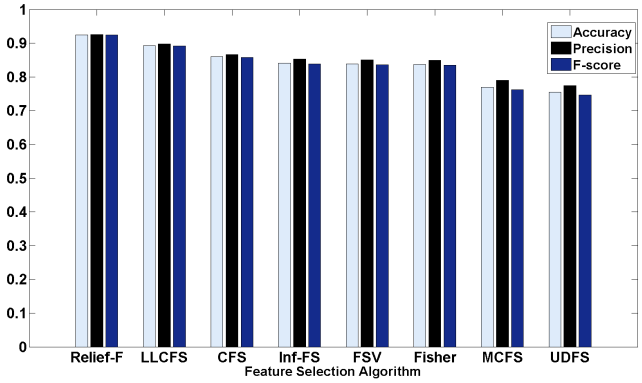

Figure 5: Comparison between the performance of different feature selection algorithms in EchoSafe. We evaluate random forest classifier using cross validation of 400 sonar cycle data samples. 50 out of 189 features are chosen by each algorithm. The candidate feature are the standard deviation of each time window and the MFCC of each microphone.

multilayer perceptron kernel (MLP). Figure 4 summarizes the cross-validation results from these classification methods. Random forest achieved the best performance, and it is used for the rest of the experiments.

The feature selection process has a significant impact on the performance of the classifier. Also, reducing the number of training samples a user has to provide, requires careful choosing for the fundamental features that capture the essence of the data in order to boost the overall accuracy and avoid over-fitting. Feature selection can be categorized into three classes: embedded methods, wrappers and filter methods. Embedded methods introduce a penalty to the complexity in order to decrease the degree of over-fitting. It also injects the selection process into the learning process. The greedy search algorithm is an example of wrappers, just to give the reader an intuition to the concept of wrappers which use classifiers to rank a subset of candidate features. FSV [2, 8] is an another example of wrappers methods. Finally, filter methods, such as Inf-FS [17, 19] and Fisher [9], analyze intrinsic properties of data, like variance for instance, and they generally ignore the classifier decision. Relief-F is an iterative, randomized, and supervised algorithm which was inspired by instance-based learning. It relies on a statistical method, few heuristics to assign weights for the candidate features. Relief-F is less often fooled, as it estimates the features ranking according to the capability of differentiating data samples that are near to each other. Furthermore, we can also categorize the feature selection algorithms into supervised and unsupervised methods. LLC-FS [27], UDFS [26], MCFS and CFS [11] are examples of unsupervised feature selection methods. Relief-F, Fisher [9] and FSV [2, 8] are under the umbrella of supervised method.

Figure 5 reports the comparison of different feature selection algorithms in terms of accuracy, precision, and F-score. We set the number of features to 50 and number of time of windows to 8. Interestingly, CFS [11] is an unsupervised method and it achieved rank three in comparison with other supervised methods. Relief-F achieves the best performance, and we use it for the rest of the experiments. CFS sorts features according to pairwise correlations. So, one could explore the unsupervised learning track for the person detection with the aid of CFS.

We also analyzed the effect of number of features in the learning process. Figure 6 summarizes the effect of increasing number of the used features on cross validation accuracy. This comparison is conducted by using 8 time windows. Relief-F is set to choose the top ranked features. Using 70 features results in accuracy of 92.87% compared with 92.45% for 50 features. EchoSafe sets 50 as a reasonable number of relevant features. Our analysis between different algorithms made use of the efforts in [18, 20].

Furthermore, we analyze the effect of dividing the whole time series data of the microphones into narrower time windows. Recall that each microphone signal is 4 seconds long and we it into non-overlapping time windows. We are using 50 relevant features using Relief-F. Figure 7 summarizes the effect of number of time windows on classification performance using cross validation. The accuracy reaches its peak of 93.13% using 32 time windows. Interestingly, the accuracy drops back to 89.25% when we use 64 time windows. Extensively dividing the data to narrower time windows loses important features of the reflected audio signals.

We performed a separate set of experiments to determine if the position of the user with respect to the smart speaker affects our results. We systematically positioned the user in four different directions of the experiment area, and collected data from 50 sonar cycles in each direction. We performed the same cross-validation based training and testing on Random Forest classifier with 50 features selected by Relief-F. The classifier accuracy was robust at 93%. Hence, EchoSafe works well regardless of the direction of the user with respect to the speaker.

6 RELATED WORK

In the following section, we compare our work with previous research in voice command attacks and defenses.

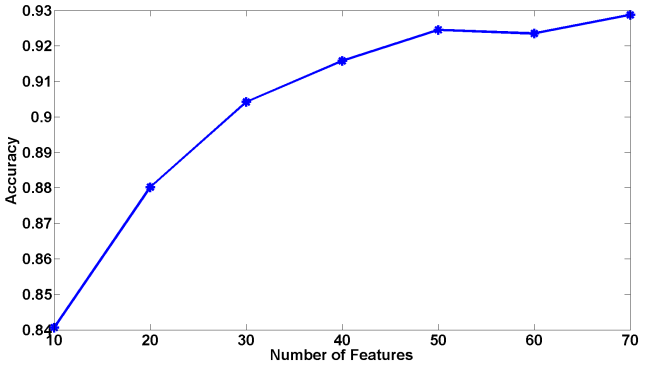

Figure 6: Analyzing the effect of number of the features in the cross validation accuracy. We use 8 time windows on each microphone audio signal, the Relief-F algorithm to select features, and Random Forest for classification.

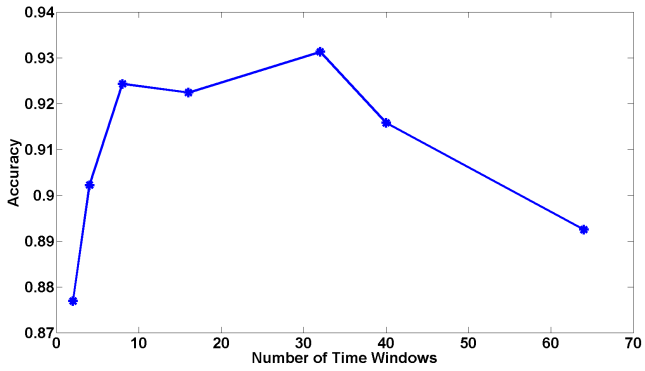

Figure 7: Analyzing the effect of number of the time windows in the cross validation accuracy. Each sonar cycle sample has 4 seconds of audio data for each microphone. We use Relief-F to select the most relevant 50 features and Random Forest for classification.

6.1 Voice command attacks and defenses

Attacks exploiting voice command interfaces enable attackers to steal sensitive information and control the devices. Researchers have demonstrated the impacts of these attacks. Diao et al. propose and implement attacks that utilize the voice command interface to steal information and take control of users' phones [5]. Carlini et al. propose attacks that are more stealthy by building voice commands that will not be heard by the users but can be processed by the machine learning applications running on the speakers [3].

To protect user against these attacks, researchers propose various defenses. However, the state-of-art defenses are clumsy. For example, there are defenses trying to identify human voice from machine voice, however, the accuracy is not high and is worse with noise [3]. Developers can introduce audio captcha as a defense to differentiate human from machine, however, the audio captcha can be easily broken by modern machine learning technology [3]. Furthermore, developers can use voice signature to identify whether the voice command is from the user or different people [15]. However, this method cannot identify attacks if attackers reuse the user's voice to create commands. Challenge response is also helpful to identify whether there is a user speaking to the device or a remote attacker playing some voice commands to fool the device. However, challenge response involves lots of user interactions. Similarly, two-factor authentication is not suitable because of usability issues [4]. Our solution is transparent to users and is effective for most scenarios of attackers.

6.2 Occupancy detection

Occupancy detection has been exploited by researchers using different sensors. Lu et al. utilize wireless motion sensors and door sensors to detect whether there is someone at home, so that they can turn the air conditioner off automatically [14]. Also, Shih et al. propose using ultrasonic chirps to detect the number of people in the room [23]. Researchers also combine multiple sensors to get an accurate estimation of the occupancy [10]. In our current solution, we use sonar to detect occupancy because we do not need to deploy extra sensors in the smarthome scenario.

6.3 Analysis using sonar

Sonar is a useful technology for many use cases. Nandakumar et al. explore active sonar to track users' fingers [16]. Dokmanic et al. use the acoustic information to generate the shape of rooms [6]. In this project, we use sonar to detect the occupancy detection, but we can also extend the approach to improve the system, such as estimating the distance of the user, or even recognize the user based on sonar signatures.

7 DISCUSSION AND FUTURE WORK

With EchoSafe, we have shown that sonar can be used to verify the presence of the user using machine learning classifiers. The results are robust to ambient noise, movement of environmental objects and the direction at which the user is present. This proof of concept shows that EchoSafe can be used as an active defense mechanism against audio attacks on smart speakers without affecting it's usability. This direction of research can be extended in various ways, to create defenses against voice attacks and in other application domains such as occupancy detection.

7.1 Limitations

EchoSafe should be improved to support non-line-of-sight interactions with the smart speaker. Also, the number of samples used for training the EchoSafe classifier should be reduced. The frequency of the sound pulse used in sonar can be moved to ultrasound, so that users are not perturbed by them. We can also send multiple randomized short pulses to improve robustness and defend against stronger attacks. If the attacker crafts the voice commands and sends them stealthy, as a noise for example, while the user is presented, then it is challenging for our current implementation to prevent such attacks. However, we can extend our sonar analysis to detect the sound source in order to prevent these crafted attacks.

7.2 Physics Based Sonar

Instead of relying on machine learning classifiers, we can implement physics based methods to robustly compare the sound coming

direction and the user standing direction. These can help with multi-person interactions and stealthy attacks. We can even capture fine grained motions such as gestures with sonar [16]. Secret gestures can be used as authentication instead of just verification of user presence. Moreover, gesture captchas can be used to thwart machine learning attacks. Gesture based interactions can also add to the usability of smart speakers.

7.3 Activity Recognition Applications

Motion sensors have long been used for occupancy based applications [14, 22], but they are not robust and can be triggered due to sunlight, wind or other thermal variations [1]. EchoSafe can be used as a robust method to detect occupancy for these applications. With physics based sonar, we can also develop methods to identify user activities, such as cooking, watching television, sleeping, etc. Then, use it for many ubiquitous computing applications.

7.4 Privacy Implications

In our threat model, we assumed that the device vendor is trustworthy. However, they have an incentive to share information about the user to third party advertisers. Amazon Echo may already store all the audio it listens to, and many different private inferences can be made from it [12]. Sonar based methods are added to these inferences, and presented a genuine threat to user privacy. Smartphone vendors have adopted several policies to limit exploitation of phone sensor data (e.g. location). We need similar strategies for smart speakers, and this presents a promising direction for future work.

8 CONCLUSION

Smart speakers are becoming commonplace in modern homes. Moreover, they are vulnerable to audio vector attacks, as they are always listening for commands. An attacker can send malicious commands to the speaker when the room is unoccupied and compromise user privacy, safety and security. In this paper, we propose a promising defense against voice command attack by verifying the presence of the user, when the command is received by the smart speaker. We design the defense utilizing sonar to detect the occupancy of the room in order to prevent remote attacks. We show that user presence can be detected using sonar with 93% accuracy, and our method is robust to ambient noise and user position in the room.

ACKNOWLEDGMENTS

This research is funded in part by the National Science Foundation under awards # CNS-1329755 and CNS-1705135, and by the King Abdullah University of Science and Technology (KAUST) through its Sensor Innovation research program. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the funding agencies.

REFERENCES

[1] Yuvraj Agarwal, Bharathan Balaji, Rajesh Gupta, Jacob Lyles, Michael Wei, and Thomas Weng. 2010. Occupancy-driven energy management for smart building automation. In *Proceedings of the 2nd ACM workshop on embedded sensing systems for energy-efficiency in building*. ACM, 1–6.

[2] Paul S Bradley and Olvi L Mangasarian. 1998. Feature selection via concave minimization and support vector machines.. In *ICML*, Vol. 98. 82–90.

[3] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. 2016. Hidden Voice Commands.. In *USENIX Security Symposium*. 513–530.

[4] CNET. 2017. Two-factor authentications. (2017). Retrieved July 27, 2017 from https://www.cnet.com/news/two-factor-authentication-what-you-need-to-know-faq/

[5] Wenrui Diao, Xiangyu Liu, Zhe Zhou, and Kehuan Zhang. 2014. Your voice assistant is mine: How to abuse speakers to steal information and control your phone. In *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*. ACM, 63–74.

[6] Ivan Dokmanić, Reza Parhizkar, Andreas Walther, Yue M Lu, and Martin Vetterli. 2013. Acoustic echoes reveal room shape. *Proceedings of the National Academy of Sciences* 110, 30 (2013), 12186–12191.

[7] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. *International Conference on Learning Representations (ICLR)* (2015).

[8] Guillermo L Grinblat, Javier Izetta, and Pablo M Granitto. 2010. Svm based feature selection: Why are we using the dual?. In *IBERAMIA*. Springer, 413–422.

[9] Quanquan Gu, Zhenhui Li, and Jiawei Han. 2012. Generalized fisher score for feature selection. *arXiv preprint arXiv:1202.3725* (2012).

[10] Ebenezer Hailemariam, Rhys Goldstein, Ramtin Attar, and Azam Khan. 2011. Real-time occupancy detection using decision trees with multiple sensor types. In *Proceedings of the 2011 Symposium on Simulation for Architecture and Urban Design*. Society for Computer Simulation International, 141–148.

[11] Mark Andrew Hall. 1999. Correlation-based feature selection for machine learning. (1999).

[12] Gierad Laput, Yang Zhang, and Chris Harrison. 2017. Synthetic Sensors: Towards General-Purpose Sensing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 3986–3999.

[13] Huan Liu and Hiroshi Motoda. 2007. *Computational methods of feature selection*. CRC Press.

[14] Jiakang Lu, Tamim Sookoor, Vijay Srinivasan, Ge Gao, Brian Holben, John Stankovic, Eric Field, and Kamin Whitehouse. 2010. The smart thermostat: using occupancy sensors to save energy in homes. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*. ACM, 211–224.

[15] Microsoft. 2017. Speaker Recognition API. (2017). Retrieved July 27, 2017 from http://www.mckinsey.com/spContent/connected_homes/index.html

[16] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota. 2016. Fingerio: Using active sonar for fine-grained finger tracking. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 1515–1525.

[17] Giorgio Roffo and Simone Melzi. 2016. Online Feature Selection for Visual Tracking.. In *BMVC*.

[18] Giorgio Roffo and Simone Melzi. 2017. *Ranking to Learn: Feature Ranking and Selection via Eigenvector Centrality*. Springer International Publishing, Cham, 19–35. https://doi.org/10.1007/978-3-319-61461-8_2

[19] Giorgio Roffo, Simone Melzi, and Marco Cristani. 2015. Infinite feature selection. In *Proceedings of the IEEE International Conference on Computer Vision*. 4202–4210.

[20] G. Roffo, S. Melzi, and M. Cristani. 2015. Infinite Feature Selection. In *2015 IEEE International Conference on Computer Vision (ICCV)*. 4202–4210. https://doi.org/10.1109/ICCV.2015.478

[21] Nirupam Roy, Haitham Hassanieh, and Romit Roy Choudhury. 2017. Backdoor: Making microphones hear inaudible sounds. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2–14.

[22] James Scott, AJ Bernheim Brush, John Krumm, Brian Meyers, Michael Hazas, Stephen Hodges, and Nicolas Villar. 2011. PreHeat: controlling home heating using occupancy prediction. In *Proceedings of the 13th international conference on Ubiquitous computing*. ACM, 281–290.

[23] Oliver Shih and Anthony Rowe. 2015. Occupancy estimation using ultrasonic chirps. In *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems*. ACM, 149–158.

[24] The Seattle Times. 2017. Amazon has sold more than 11 million Echo devices, Morgan Stanley says. (Jan. 2017). Retrieved July 27, 2017 from http://www.seattletimes.com/business/amazon/amazon-has-sold-more-than-11-million-echo-devices-morgan-stanley-says/

[25] The Verge. 2017. Amazon's Alexa started ordering people dollhouses after hearing its name on TV. (Jan. 2017). Retrieved July 27, 2017 from https://www.theverge.com/2017/1/7/14200210/amazon-alexa-tech-news-anchor-order-dollhouse

[26] Yi Yang, Heng Tao Shen, Zhigang Ma, Zi Huang, and Xiaofang Zhou. 2011. l2, 1-norm regularized discriminative feature selection for unsupervised learning. In *IJCAI proceedings-international joint conference on artificial intelligence*, Vol. 22. 1589.

[27] Hong Zeng and Yiu-ming Cheung. 2011. Feature selection and kernel learning for local learning-based clustering. *IEEE transactions on pattern analysis and machine intelligence* 33, 8 (2011), 1532–1547.