Acoustic Keylogger Report

Aviad Shalom Tzemah
Yuval Levy

28.06.2022

1 Introduction

Side-channel attacks are a family of attacks utilizing any extra obtainable information from a computer protocol or algorithm implementation, rather than from flaws in its design, this includes information like power consumption, electromagnetic leaks and sound. Keylogging is an example of such an attack. It is the practice of recording the keys a person types leveraging emanations from a keyboard. While this itself is legal, with applications in studying human-computer interaction and allowing employees to oversee the use of their computers, it may also serve for security intrusions at the hands of malicious users. Indeed, Keylogging side-channel attacks have become a topic of significant interest over the years, with the first attack discovered over five decades ago. Keylogging may exploit electromagnetic wave emitted from electrical components in the keyboard [6], vibrations of the surface under the keyboard caused by typing [5] and even CPU cache usage [3].

One area in which the research community has invested a lot of effort in studying is keyboard acoustic emanations, demonstrating its privacy risk to inferring sensitive data. The main observation, and the reason why keyboard acoustic emanations leak information, is that different keys on the keyboard make different click sounds. Efforts in this field can be divided based on whether they use statistical properties of the sound spectrum or timing information. For example, one convenient way to exploit timing information is using multiple microphones and analyze the Time Difference of Arrival (TDoA) information to triangulate the position of the pressed key [4]. Approaches that use statistical properties of the sound spectrum typically apply machine learning.

One insight seen in [7] was that the typed text is often not random. When one types English text, for instance, the finite number of mostly used English words limits possible temporal combinations of keys, and English grammar limits word combinations. Using unsupervised approaches one can first cluster keystrokes into a number of acoustic classes based on their sound, then given sufficient (unlabeled) training samples, a most-likely mapping between these acoustic classes and actual typed characters can be established using the language constraints. Another finding appearing in [2], is that the keystrokes sounds make correlate to their physical positioning on the board, making it possible to generate sets of constraints from recorded sounds and select words from a dictionary that match these constraints.

The supervised learning approaches, like in [1], depend on having previously recorded the acoustics of a specific keyboard, where individual keys can generally be identified by comparing the unknown waveforms to the known waveforms collected during training. Good performance can be achieved in eavesdropping on the input to the same keyboard, or keyboards of the same model. Authors use Fast Fourier Transform (FFT) coefficients of the audio signal as features and a neural network to recover text that can also be random.

Notwithstanding the aforementioned methods, previous attacks often assumed assumptions that are not very practical in many real world settings. Such include adversary's physical proximity to the victim, precise profiling of the victim's typing style and keyboard, and/or significant amount of victim's typed information (and its corresponding sounds) available to the adversary. In this chosen paper, authors present a new keyboard acoustic attack - **Skype & Type** (S&T) - that avoids prior strong adversary assumptions, made through eavesdropping to Voice-over-IP (VoIP) calls - most specifically via Skype. This is made possible on the basis of the hypothesis that often people engage in secondary activities during these kind of calls. Activities such as writing email, for example, involve using the keyboard. As emanations are picked up easily with the VoIP software, it possible to determine what the user typed based on keystroke sounds, should a malicious party is part of the conversation. This is innovative since prior studies have not considered either the setting of the proposed attack, or the features of VoIP software.

2 Method

Prior to describing the attack settings, the following assumptions are made: (1) target-device has VoIP software installed and a built-in or attached keyboard, (2) the only acoustic information the attacker receives from the victim is transmitted by the VoIP software, (3) if any *ground truth* * is disclosed it is small, (4) target-text is very short and random, corresponding to an ideal password. S&T attack operates under 3 scenarios:

- Complete Profiling: The attacker knows some of the victim's keyboard acoustic emanations on target-device, along with the ground truth for these emanations, i.e. possesses some labeled data. Under this scenario the attacker has maximum information about the victim.

- User Profiling: The attacker has no labeled data, however has prior knowledge about the keyboard model. Training data is obtainable during the call, thus the victim's typing style could be profiled.

*Some keyboard sounds together with the corresponding text. Ground truth could also be collected offline, if the attacker happened to be near the victim, at some point before or after the actual attack.

- Model Profiling: In this setting, the attacker has no information about the keyboard, nor possession of training data. Hence, keyboard acoustic emanations has to be first be used to classify the target device, this is done using a database of sounds from previous attacks.

The attack workflow is the same for all settings, with a data processing stage and a classification stage. Of note, under the Model Profiling scenario, the classification phase is nested, comprised of first identifying the target-model, and only then classifying the keys pressed:

1. **Data Segmentation** - Intended to isolate distinct keystrokes within the recording. Since pressing sounds are generally louder than releasing, only the former sound peaks were considered. This was done by normalizing amplitudes of the signals to have a root mean square of 1. Then, FFT coefficients were summed over windows of 10ms. Whenever the energy of a windows suppresses a certain (tunable) threshold, a press event is recorded. Past an event, subsequent 100ms were taken as the waveform.

2. **Feature Extraction** - The mel-frequency cepstral coefficients (MFCC) were extracted to use as features.

3. **Classification** - As mentioned above, target-device classification refers to the Model Profiling scenario alone. Only when the attacker profiles the victim on a target device, could key classification be carried out. A k-NN classifier was used with $k = 10$ to preform the former task, while logistic regression was the model of choice for the 26-class classification of the keys.

3 Experiments

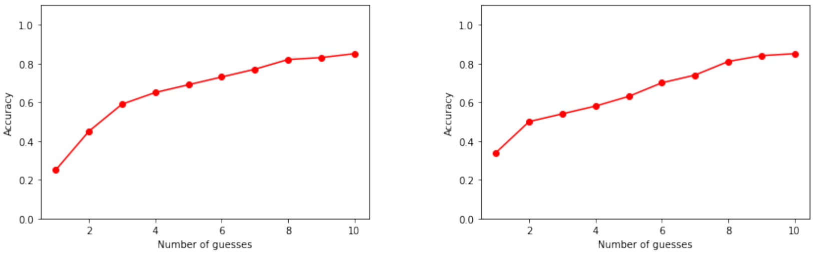
We have conducted several experiments testing the attack on different settings and conditions as follows:

3.1 Laptop attacks

We first tested the attack on a laptop, an Asus GL502-VS. We trained a model on a train set containing a sample of every letter of the English alphabet (A through Z) 10 times and 10 samples of space for a total of 270 samples.

3.1.1 Different settings of getting the attack recordings

We wanted the check how well the model will predict the recorded input when the model was trained on recordings of one setting and the attack happened in a different setting. In this case, the attack happened in a different room from where the model was trained on. As we can see in Figure 1 there is a slight decline in accuracy when compared to the same attack which happened in the same setting as the training (0.54 average accuracy for top-3 guesses



(a) Same setting as training

(b) Different setting from training

Figure 1: How well the model predicts the recorded input in different settings

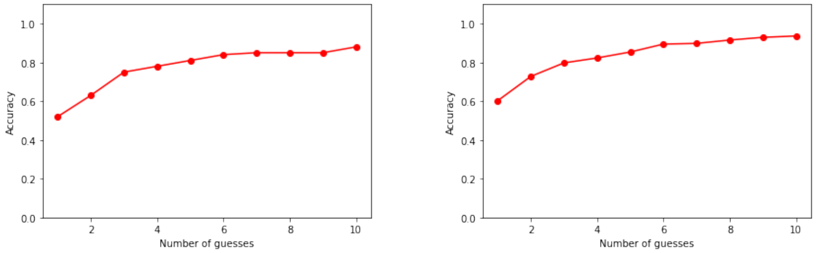
as opposed to 0.59, and 0.63 average accuracy for top-5 guesses as opposed to 0.69). Although we can observe that for the top-1 accuracy the model predicted better for the input recorded in the different setting (0.33 average accuracy as opposed to 0.25). We can conclude that the attack can generalize to different settings and is not limited for setting in which the training data was acquired in.

3.1.2 Realistic training set

As can be expected, when an attacker acquires the training set of recordings of a victim, he is going to acquire recordings of texts in a natural language (In our case, English) which means the frequency of each input (letters, spaces, symbols etc.) is not going to be the same. We trained a model on a training set which tries to imitate the frequency of the English alphabet and spaces. As we expected, when this model tries to predict a random sequence of letters it is having a bit more difficulty when compared to predicting of recordings of sentences as can be seen in Figure 2 (0.75 average accuracy for the top-3 guesses as opposed to 0.79, and 0.81 average accuracy for the top-5 guesses as opposed to 0.85). This can be explained by the fact that the recordings of sentences represent better the distribution of letters found in the language which according to this fact the model was trained on. Although, the average accuracy on the random strings of letters (which can correspond to attacking a recording of a password being typed) still works quite well with average accuracy of 0.52 at top-1 guesses and up to 0.88 average accuracy at top-10 guesses.

3.2 Desktop attack

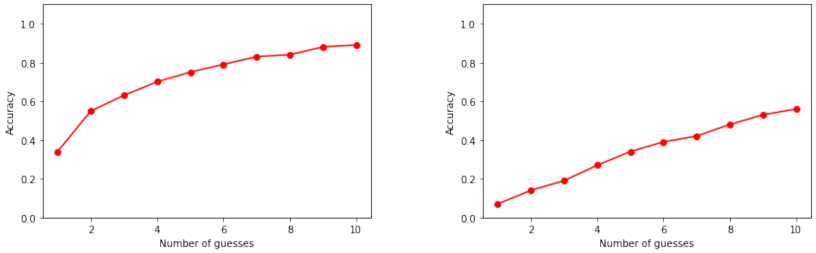
Our main premise when testing on the laptop case was that the microphone is always at a fixed distance from the keyboard. We wanted to test the attack on the case where the victim is using a desktop computer where the microphone and keyboard are not always at a fixed distance from each other. For the keyboard we used a Keychron Q1 with Gateron Phantom Red switches and for the microphone we used the built-in michrophone of a Bose QuietComfort 35



(a) Attack on a string of random letters

(b) Attack on a real sentence

Figure 2: How well a realistic model predicts different types of recordings



(a) Attack on a string of random letters, same microphone position as training

(b) Attack on a string of random letters, different microphone position than training

Figure 3: How well a realistic model predicts different types of recordings

II.

3.2.1 Microphone at different positions

Since the microphone is not always at fixed position in this case we wanted to see how would that affect the predictability of the model. Figure 3 shows the difference of the predictability of a model trained on all of the letters of the English alphabet and tested on a recordings of random strings of letters in two different microphone positions.

As opposed to the laptop case where we saw a slight decline in the average accuracy of the model over the different number of guesses, in this case we can see a huge decline in performance, 0.07 average accuracy of the top-1 guesses as opposed to 0.34, 0.19 average accuracy of the top-3 guesses as opposed to 0.63 and 0.34 average accuracy of the top-5 guesses as opposed to 0.75. We can understand from that that the most important setting for the attack is the position of the microphone relative to the keyboard. In the laptop case, even though we changed the setting of the attack the relative position of the microphone to the keyboard stayed the same and we only saw a slight decline in performance. But in the desktop case, we only changed the position of the microphone relative to the keyboard (while everything else stayed the same) and we saw a substantial decrease in performance.

4 Discussion

We think that acoustic keyloggers are a viable threat everyone should worry about. We saw that the attack works quite well on laptop computers because of the fixed position of the microphone relative to the keyboard. But doesn't quite work when the victim can move around his microphone. Also, a model trained on a natural language letters' frequency doesn't affect much the performance of the model. Furthermore, We saw that chaining the attack with a spell checker can make the life of the attacker easier when trying to understand what the victim wrote. And we think if the attacker would use more sophisticated NLP models to try and figure out what the victim wrote it might yield even better results, but further research is required.

References

[1] Dmitri Asonov and Rakesh Agrawal. Keyboard acoustic emanations. In *IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004*, pages 3–11. IEEE, 2004.

[2] Yigael Berger, Avishai Wool, and Arie Yeredor. Dictionary attacks using keyboard acoustic emanations. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 245–254, 2006.

[3] Daniel Gruss, Raphael Spreitzer, and Stefan Mangard. Cache template attacks: Automating attacks on inclusive {Last-Level} caches. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 897–912, 2015.

[4] Jian Liu, Yan Wang, Gorkem Kar, Yingying Chen, Jie Yang, and Marco Gruteser. Snooping keystrokes with mm-level audio ranging on a single phone. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 142–154, 2015.

[5] Philip Marquardt, Arunabh Verma, Henry Carter, and Patrick Traynor. (sp) iphone: Decoding vibrations from nearby keyboards using mobile phone accelerometers. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 551–562, 2011.

[6] Martin Vuagnoux and Sylvain Pasini. Compromising electromagnetic emanations of wired and wireless keyboards. In *USENIX security symposium*, volume 1, 2009.

[7] Li Zhuang, Feng Zhou, and J Doug Tygar. Keyboard acoustic emanations revisited. *ACM Transactions on Information and System Security (TISSEC)*, 13(1):1–26, 2009.