AI-based Sound-Squatting Attack Made Possible

# AI-based Sound-Squatting Attack Made Possible

Rodolfo Valentim
*Politecnico di Torino*
*Turin, Italy*
*rodolfo.vieira@polito.it*

Idilio Drago
*Università degli Studi di Torino*
*Turin, Italy*
*idilio.drago@unito.it*

Federico Cerutti
*Università degli Studi di Brescia*
*Turin, Italy*
*federico.cerutti@unibs.it*

Marco Mellia
*Politecnico di Torino*
*Turin, Italy*
*marco.mellia@polito.it*

***Abstract*—Domain squatting is an efficient attacking technique that relies on the similarity between domain names to trick users. Sound-squatting is a type of domain squatting that explores the similarity in the pronunciation of domains. Sound-squatting requires better approaches to protect users, and indeed it demands more research attention due to popularization of intelligent speakers and the increase of voice-based navigation. In this work we propose an AI-based methodology to automatically build sound-squatting candidates. We leverage recent results of AI, namely the ability to translate text, to automatically generate possible sound-squatting candidates. We evaluate our methodology by verifying the generated candidates and classifying them according to their threat class. We generate over twenty thousand candidates from popular domains, out of which, 7% are found active at the time of the analysis. Active domains include "Parked/Ads/For-Sale" domains. We thus show that automatic sound-squatting generation is useful to proactively check and limit the abuse of such offences.**

***Index Terms*—squatting, transformers, proactive security, deception for offense**

## 1. Introduction

A consequence of the Internet ubiquity is the increase in ways to profit with scams and the stealing of users' data. The constant update of security tools and measures needed to keep up with the advance of cyber threats requires efforts to anticipate new attacking techniques. One common attack is domain squatting, which occurs when attackers register perceptively confusing domain names aiming at tricking visitors into them [1]. There are several types of domain squatting: typo-squatting, bit-squatting, homograph-squatting, sound-squatting, combo-squatting [1] and skill-squatting [2]. Each type explores one perception aspect to luring users into the false domains.

Among these techniques, sound-squatting has received little attention [1], while gaining traction with the advent of smart speakers and voice-assistants [2]. Sound-squatting explores the pronunciation of domains and the fact that different words might present the same sound even if written differently. Sound-squatting is challenging because pronunciation varies from language to language and user to user, which may introduce different errors when typing or speaking a word. So far, the state-of-art proposes the production of sound-squatting candidates relying on statically built lists of homophones, which however is hard to generalize.

*The research leading to these results has been funded by the Huawei R&D Center (France) and the SmartData@PoliTO center for Big Data technologies.*

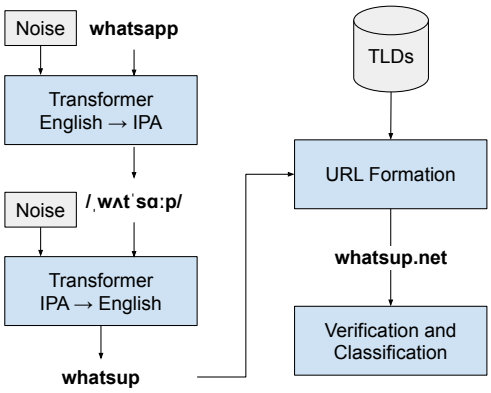



Figure 1: Methodology to generate and verify sound-squatting candidates.

We hypothesize that a data-driven approach is capable of producing more comprehensive sound-squatting candidates. To investigate this hypothesis, we leverage recent advances of AI and, in particular, models used to translate texts. Our proposal is summarized in Fig. 1. It leverages a data-driven methodology based on the Transformer Neural Network to translate English words to pronunciation and vice-versa, encoding sounds using the International Phonetic Alphabet (IPA). As opposed to Transformer Networks used for translations, our model is customized for variability by adding random noise to the latent representation of the encoder. This step allows us to recover domains which are phonetically similar to the original ones, thus increasing the number of homophones found for the same domain.

We generate sound-squatting candidates for popular second-level names and combine them with different top-level domains. After the generation we verify the inference by looking up if the domains are active. We classify the active domains using the threat classes proposed in [3].

We select 273 popular second-level domains (in the Top-500 Alexa) for evaluating our method. The model automatically generates more than 20 000 unique domains, among which 1 339 were found registered and active. Around 45% of these are ' "Parked/Ads/For-Sale" domains, and can be potentially abused by attackers later.

We show that the intersection between the candidates generated by our method and a previous work [3] is small, while ours allows the generation of more squatting candidates.

We believe our work catalyzes a proactively search for sound-squatting vulnerabilities that could be used to attack specific high profile domains. Moreover, our data-driven approach is suited to handle other languages, and can be extended to support language specific and cross-language attacks.

The rest of the work is organized in the following way: we present some prior knowledge required to understand our proposal at Section 2. Our proposal is detailed in Section 3. Section 4 presents our results, while Section 5 presents concluding remarks and our future work.

## 2. Background

### 2.1. Sound-squatting

Since its discovery in 2014 the relevance of sound-squatting has increased due to the popularization of intelligent speakers, which heavily rely on voice commands to access apps and services. Some works have already explored this angle and proposed solutions to increase the security for users [2], [9]. The use of pronunciation similarity for scams has been a concern not only for smart-speaker users, but also for visually impaired people [4].

However important, this type of attack has been left out of state-of-art works proactively searching for squatting [5]. The attack replaces portions or the complete domain name by a word with a similar pronunciation. An user, when idly reading the link or using voice navigation systems, might confuse the squatting domain for the real one. Sound-squatting is a type of attack that varies depending on the person speaking or reading, the level of skill in the language and the medium used to navigate the web. Therefore, sound-squatting is more effective when mitigation methods rely solely on statically and manually built lists of homophones (as in [3]), which are naturally limited by their creators' intuition.

### 2.2. Transformers Neural Network

Recent advances in AI and Natural Language Processing (NLP) have made it practical to automate text translation via sequence-to-sequence (seq2seq) mapping. The Transformer model [6] represents a step forward. Compared to other seq2seq models such as Recurrent Neural Network (RNN), the Transformer has the capability of working with large sequences without losing information. This is possible thanks to the use of the attention mechanism that learns the relationship between all elements in sequences. The attention mechanism is analogous to a retrieval system, where there is a Query (Q), a Key (K) and a Value (V). Roughly, Q represents the word and the K:V represents the memory. The attention works like a database system where we query the memory, compare our query with a set of keys and get the corresponding values. In a Transformer, the attention mechanism is applied to the input sequence of the translation, to the target language and to the ongoing merge of the source sequence with its translation.

Several Transformer variants exist, not all restricted to NLP problems. Examples of algorithms include BERT [7] and GPT-2 [8]. Transformers have achieved the highest scores for Neural Machine Translation [6], [12] and their performance has represented a jump in quality for seq2seq problems.

### 2.3. International Phonetic Alphabet

The International Phonetic Alphabet (IPA) is widely used for representing pronunciation. The International Phonetic Association maintains a consistent method to represent sounds in written form. IPA contains a unique character for each sound and it is capable of representing intonation and other properties of any language.

The IPA alphabet has changed over time to reflect new discoveries in linguistics. It is stable and representative enough to be used in Machine Learning applications. There exist dictionaries with IPA pronunciation for several different languages, including English [13], which we use in this work.

## 3. Methodology

We now detail our methodology for generating sound-squatting, as well as how we verify whether the generated domains represent active threats.

### 3.1. Solution Overview

We employ Transformer models to translate from English to IPA and from IPA back to English. We fed the first Transformer with the sequence of characters in a word (i.e., the target domain) to obtain as intermediate output the sequence of IPA tokens. The second Transformer then performs the inverse operation. By linking the English-to-IPA and the IPA-to-English translators, as shown in Fig. 1, we close the loop and reconstruct the input word. We train our models using a large dataset of English words and their pronunciations [13].

With well trained models, the expected behavior when submitting an English word to the composite model is to obtain the same given word after the English-to-IPA and the IPA-to-English translations. In this scenario, there could be inconsistencies between input and output only for homophones, where a word shares the IPA pronunciation with other words.

As we want to enforce the generation of homophones as well as words with slightly different pronunciation, we add noise to the latent representation of the input in our translation model. This scheme is depicted in Fig. 2. We add noise both during the training and during the inference.

We add noise at the training phase by summing a random value to the latent representation of the encoder (see Fig. 2). This is required because the encoder does not map the input to a distribution and small changes in the encoder output do not result in small changes in the output. Therefore training the model with the noise teaches it to handle small variations.

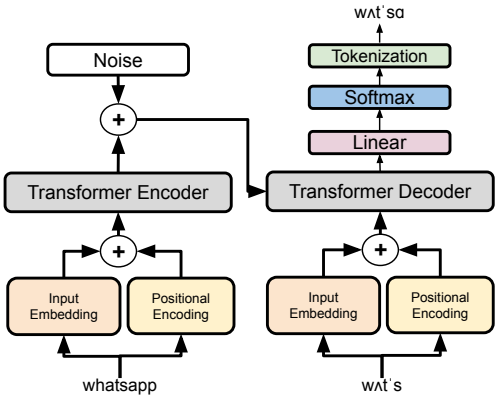



Figure 2: Block diagram for our Transformer architecture.

### 3.2. Sound-Squatting Generation

We obtain a list of sound-squatting candidates for each target domain. We first train our Transformer models capable of translating from English to IPA and from IPA to English, using the architecture in Fig. 1 and our training dataset [13].

We then collect popular domains, which intuitively, are more likely to be a target for identity-based attacks. We use the Alexa Top-500 ranking as the source for selecting our seed domains. We select only the second-level domains for those domains in the Alexa Top-500, e.g., getting `example` for the `example.com` entry. Subsequently, we remove domains that are simple acronyms (e.g., `abc.com`), because our model has not been trained to handle the pronunciation of acronyms. This selection stage has resulted in 273 unique domains out of our initial list of 500 candidates.

With this initial list of domains we generate candidates for each name (e.g., `exemple` and `exempal`). We repetitively give the model the same input and, at each run, receive possibly different outputs due to the additional injection of noise. After a fixed number of attempts per domain (10 times) we move to the next name in the list. This procedure produces a different number of candidates per input domain, as the Transformer might generate repetitions. At the end, we remove the duplicates and combine the generated names with Top-level domains to compose sound-squatting candidates.

### 3.3. Verification

The verification is divided into two stages. In the first stage we consider only if the domain is active by checking whether there is an IP address associated with the sound-squatting candidate. Knowing if a domain is active shows some interest in the name, which could host an attack in the future. We then also remove inactive domains that would decrease the speed of the second verification.

The second verification is time consuming and involves a manual verification to classify active domains into classes of threats. We use the same classification proposed in [3] for comparison between methods. The classes are:

**Parked/Ads/For-Sale Domains:** contains no real content, except ads which are constructed on demand, usually by a domain-parking service.

**Authoritative-Owned Domains:** managed by the companies and organizations behind the corresponding original domain used as target.

**Legit Domains:** are legit names of other companies or brands that are not deliberately abusing the similarity with the target domain.

**Hit-stealing Domains:** sound-squatting to capture traffic and feed the attackers' own "business related" domains with hits intended for the authoritative site.

**Scam-Related Domains:** Domains supporting scam attacks against users.

**Affiliated Domains:** domains that redirect to the correct page, however, attaching some reference code that indicates they are responsible for redirecting these users and, therefore, receiving commission for the redirection.

**Others:** domains that for some reason load white or error pages.

**Errors:** domains that do not serve any page, even if registered.

To support this check, we instrument a Chrome Browser with Selenium to visit all second-level candidate domains, collecting a screenshot of the page after 10 seconds. We then manually evaluate the obtained screenshots to classify the candidate domains, eventually revisiting the page to confirm our decisions if necessary.

## 4. Results

We next detail how we implemented the above methodologies and provide results.

### 4.1. Model Training and Generation

The Transformer architecture we use contains 2 heads, latent dimension 2 048 and sequence size 25. We used Keras Framework[14] to implement and we trained locally. We train the model for 10 epochs using a batch size of 64. Our training dataset has 107 038 samples and 18 889 samples form our validation dataset. After 10 epochs the English-to-IPA translator achieves an accuracy of 92.7% and the IPA-to-English, 88.4%.

The hyper-parameters have been chosen after some feature engineering. The reduced number of heads compared to other NLP setups is justified by the fact that

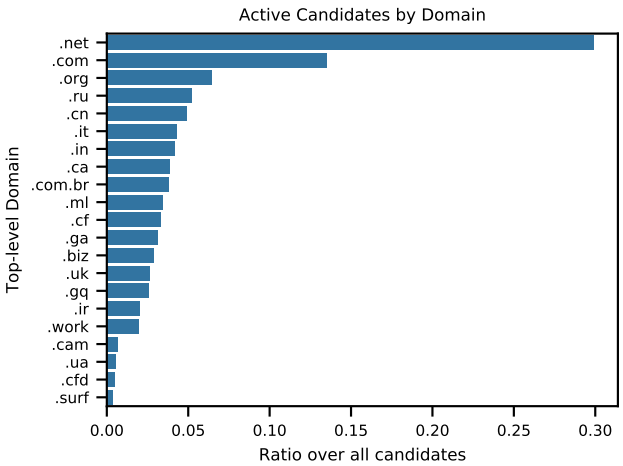
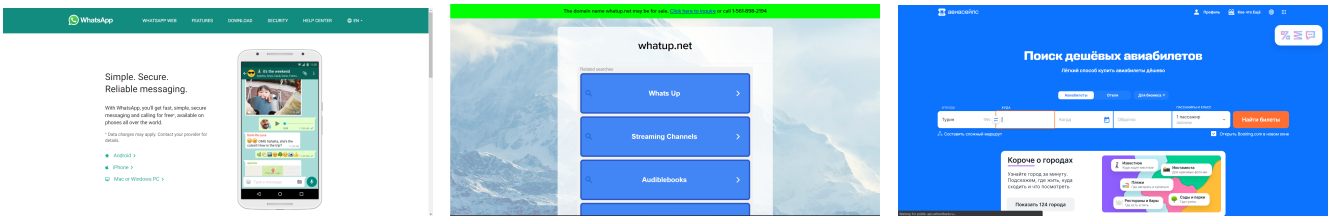



Figure 3: Active sound-squatting candidates by TLD.




(a) whatsapp.com (b) whatup.net (c) whatsup.ru

Figure 4: (a) Legit domain. (b) Parked Domain example (c) Hit-stealing Domain example.

the translation from English to IPA is a task that does not require the model to learn long dependencies in the sequence.

Recall that we start from the 500 most popular domains in the Alexa Ranking, and are left with 273 unique target domains after removing acronyms (shorter than four characters) and duplicates. As we want to generate multiple homophones, we run the generation 10 times for each target. Our Transformers also generate repetitions – in fact, for short names, there are limited alternatives for creating reasonable homophones. After running the generation step as above and removing the generated duplicates, we end up with 910 unique names.

We combine each candidate with 21 Top-level domains (TLD) to form a complete name. We select TLDs listed as the most abused by the Spamhaus Project [10], as well as other TLDs of interest to the authors, such as: net, com, org, com.br, it, ru, in, ir and others (see also Fig. 6). At the end, we obtain a list of 19 110 sound-squatting candidates to check.

Out of the 19 110 sound-squatting candidates, 1 339 candidates are active by the time of writing, which corresponds to 7% of the total. Fig. 3 show the results by breaking them down relatively to the considered TLD. We notice the most of the active candidates are concentrated at the most popular TLDs in the list, i.e., .net, .com, and .org.

## 4.2. Threat Classification

The second stage in our verification requires us to compare each screenshot with the respective legit page. This verification is necessary to qualitatively evaluate if the model generates reasonable sound-squatting candidates. Fig. 4 exemplifies the method. The original domain (whatsapp.com) has 2 active sound-squatting candidates, i.e., whatup.net and whatsup.ru: i) whatup.net domain is a *parked* domain in which we can see ads and text indicating the possibility to buy the domain; and ii) whatsup.ru is a candidate that is *hit-stealing* from whatsapp.com and redirecting users to a completely unrelated website.

We manually categorized all active candidates into the eight classes. Fig. 5 shows the results. We notice the majority of the candidates are used for "Parked/Ads/For-sale". Moreover, there is a considerable amount of Hit-stealing Domains and Authoritative Domains. The Authoritative Domains are mostly from e-commerce sites (e.g., Amazon and Shopify) that tend to protect their businesses by proactively registering possibly abused domains. Finally, we have found some Scam Domains that

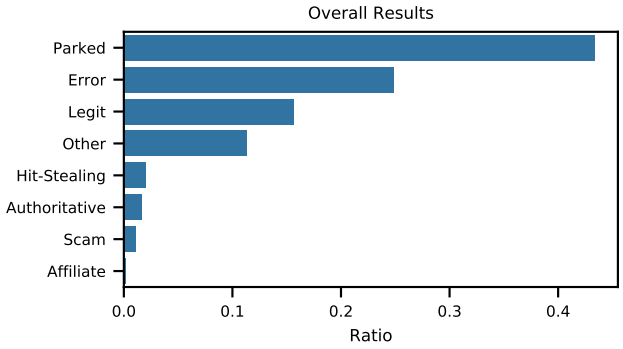



Figure 5: Overall results obtained by manual analysis of candidates screenshots compared with legit domains.

corroborate with our hypothesis that our model is capable of mimicking attackers' intentions.

Fig. 6 analyzes the threat classes regarding the TLDs. The general trend across TLDs is similar, with Parked domains representing the majority of the candidates as well as large percentages of errors, legitimate domains and other type of errors. Yet, some interesting patterns emerge. First, notice the clear differences on the percentages for malicious categories across TLDs, such as Scam and Hit-Stealing. This result somehow suggests that the abusing of sound-squatting follows general attacking trends.

Second, notice how authoritative cases are more common in some TLDs. As said above, these are the cases of businesses protecting their brands, and the figure suggests they pay more attention for particular TLDs. For TLD .in, there are 6 Authoritative Domains: amazen, amazan, flicker, shopifi, shopfy and skipe.[1] The examples suggest a tendency for authoritative domains to take actions related to specific pronunciations of the brands, such as replacing, removing or adding letters that have little affect in the pronunciation. We notice the same tendency for the TLD highly used by latinophile TLD i.e., `.com.br` and `.it`.

## 4.3. Comparison with Homophone-based Method

We now compare our proposal with the method proposed in [3], in which the authors use a manually-generated list of substrings to replace homophones in names. We start from the same list of 273 second-level domains, and generate 502 candidate domains with their

1. Out the 910 unique names, 113 can also be considered simple typo-squatting – e.g., the Edit Distance is 1 and the distance between the swiped characters in the keyboard is small. The 113 names result in 643 active domains, 50% of the found hit-stealing and around 22% of the found scam domains.

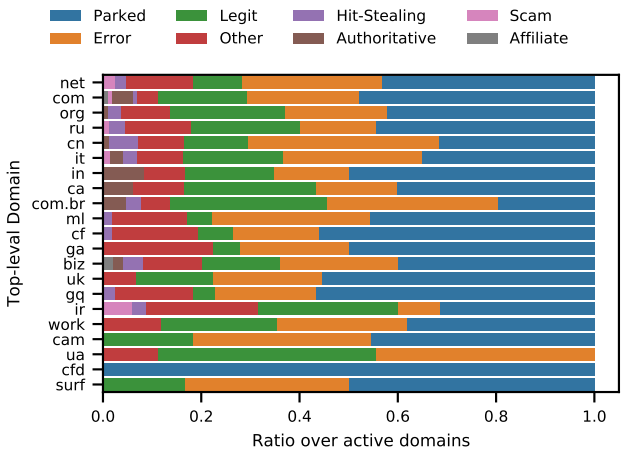



Figure 6: Ratio of each threat class per Top-level Domain.

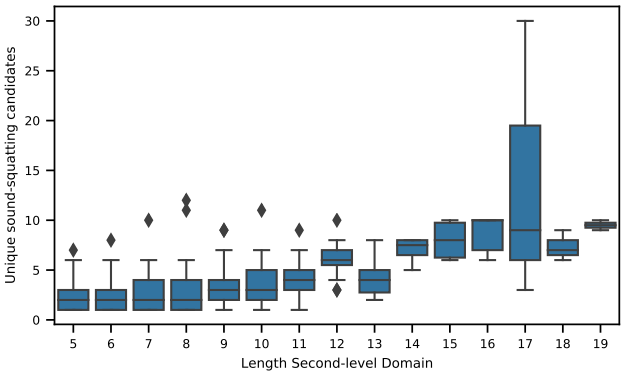



Figure 7: Unique candidates per domain length. Note that we have repetitions on the second-level domain and for this reason there are outliers bigger than ten which is the number of repetitions.

approach. Sound-squatter generates 910 candidates, and recall that we stop the generation after 10 tentative per input domain. The intersection among the two sets consists in only 6 domains (amason, blogspott, flicker, reutters, tumbler, zdnett). This suggests that the two approaches are complementary, with the advantage of our methodology to be fully automated and extensible to other languages than English. In the Table 1 we report some examples of generated sound-squatting domains.

TABLE 1: Examples of sound-squatting candidates for AI- and Homophone-based methods.

| Domain | AI-based | Homophone-based |
|---|---|---|
| facebook.com | fasbook.com | facebeuk |
| cloudflare.com | cloudflaire.net | cloudflair |
| spotify.com | spotifi.net | spottify |
| tumblr.com | tumbler.net | tumbler |
| yahoo.com | yahu.com | yahou, yahoux, yawho |
| whatsapp.com | whatsup.net | whattsapp |

Finally, to understand how many candidates we can generate and what are the dynamics in the generation, we investigate the relationship between the number of unique sound-squatting domains and the length of the original domain name. The results are shown in Fig. 7 and indicate that the bigger the string the higher is the chance to find candidates. Indeed, we see a correlation between the length and the average number of obtained sound-squatting candidates. These results corroborate the idea that some domains are more vulnerable to sound-squatting than others. They also show the flexibility of our methodology in producing larger lists of candidates than methods based on fixed lists of homophones.

## 5. Conclusion and Future Works

We presented a methodology for a data-driven sound-squatting generation to prevent the offense by proactive search. We use pronunciation data for a target language to automatically replace syllables with similar sound in a word. To achieve this, we leverage the Transformers architecture to create a translator for English to IPA and IPA to English. By adding noise in each translator, we introduce errors in the process, generating possible homophones. We used our model to generate candidates for the top-500 Alexa domains obtaining more than 20 000 possible candidates, 7% of which are found active in various malicious classes. Sound-squatter proves the capability of automatically finding domains that use the sound similarity between its name and the popular counterpart to attract traffic and possibly lure users. Interestingly, we have found companies, especially e-commerce sites, that appears to be aware of this risk and proactively registered sound-squatting candidates to protect their websites. Our proposal can help them in automating and generalizing this process, and in general it can be used to proactively find possible sound-squatting attacks to any given domain.

In future work we will extend our methodology for other languages. While including other languages requires only to train another sound-squatter translator chain, generalizing the approach to mimic possible errors a non-native speaker may introduce is not trivial. Because, different languages use different sets of sounds to form the vocabulary and phonemes, sounds are not present in all languages, which incur in gaps between phonemes and graphemes while translating. This variety challenges the training of translators. Another aspect we will work on the future is the assessment of the quality of the generation, which might involve human evaluation.

## References

[1] Y. Zeng, T. Zang, Y. Zhang, X. Chen, and Y. Wang, "A comprehensive measurement study of domain-squatting abuse," in ICC 2019-2019 IEEE International Conference on Communications (ICC), 2019, pp. 1–6.

[2] D. Kumar et al., "Skill squatting attacks on Amazon Alexa," in 27th USENIX security symposium (USENIX Security 18), 2018, pp. 33–47.

[3] N. Nikiforakis, M. Balduzzi, L. Desmet, F. Piessens, and W. Joosen, "Soundsquatting: Uncovering the use of homophones in domain squatting," in International Conference on Information Security, 2014, pp. 291–308.

[4] G. Sonowal and K. Kuppusamy, "Mmsphid: a phoneme based phishing verification model for persons with visual impairments," Information & Computer Security, 2018.

[5] K. Tian, S. T. Jan, H. Hu, D. Yao, and G. Wang, "Needle in a haystack: Tracking down elite phishing domains in the wild," in Proceedings of the Internet Measurement Conference 2018, 2018, pp. 429–442.

[6] A. Vaswani et al., “Attention is all you need,” Advances in neural information processing systems, vol. 30, 2017.

[7] I. Tenney, D. Das, and E. Pavlick, “BERT rediscovers the classical NLP pipeline,” arXiv preprint arXiv:1905.05950, 2019.

[8] A. P. B. Veyseh, V. Lai, F. Dernoncourt, and T. H. Nguyen, “Unleash GPT-2 power for event detection,” in Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics, 2021, pp. 6271–6282.

[9] N. Zhang, X. Mi, X. Feng, X. Wang, Y. Tian, and F. Qian, “Understanding and mitigating the security risks of voice-controlled third-party skills on amazon alexa and google home,” arXiv preprint arXiv:1805.01525, 2018.

[10] The Spamhaus Project. https://www.spamhaus.org/. [Online]. Available: https://www.spamhaus.org/

[11] F. Quinkert and others, “Be the Phisher - Understanding Users’ Perception of Malicious Domains,”in Proceedings of the 15th ACM Asia CCCS, 2020, pp. 263–276. doi: 10.1145/3320269.3384765.

[12] X. Liu, K. Duh, L. Liu, and J. Gao, “Very deep transformers for neural machine translation,” arXiv preprint arXiv:2008.07772, 2020.

[13] L. Doherty, ipa-dict - Monolingual wordlists with pronunciation information in IPA. GitHub, 2022. [Online]. Available: https://github.com/open-dict-data/ipa-dict

[14] F. Chollet and others, “Keras,” 2015. https://github.com/fchollet/keras