2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW) | 979-8-3503-2720-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/EuroSPW59978.2023.00063

Lost in Translation: AI-based Generator of Cross-Language Sound-squatting

Rodolfo Valentim
Politecnico di Torino
rodolfo.vieira@polito.it

Idilio Drago
Università di Torino
idilio.drago@unito.it

Marco Mellia
Politecnico di Torino
marco.mellia@polito.it

Federico Cerutti
Università degli Studi di Brescia
federico.cerutti@unibs.it

***Abstract*—Sound-squatting is a phishing attack that tricks users into accessing malicious resources by exploiting similarities in the pronunciation of words. It is an understudied threat that gains traction with the popularity of smart-speakers and the resurgence of content consumption exclusively via audio, such as podcasts. Defending against sound-squatting is complex, and existing solutions rely on manually curated lists of homophones, which limits the search to a few (and mostly existing) words only. We introduce Sound-squatter, a multi-language AI-based system that generates sound-squatting candidates for proactive defense that covers over 80% of exact homophones and further generating thousands of high-quality approximated homophones. Sound-squatter relies on a state-of-art Transformer Network to learn transliteration. We search for Sound-squatter generated cross-language sound-squatting domains over hundreds of millions of emitted TLS certificates comparing with other types of squatting candidates. Our finding reveals that around 6% of generated sound-squatting candidates have emitted TLS certificates, compared to 8% of other types of squatting candidates. We believe Sound-squatter uncovers the usage of multilingual sound-squatting phenomenon on the Internet and it is a crucial asset for proactive protection against sound-squatting.**

1. Introduction

Cyber-squatting is a phishing attack that tricks users into accessing malicious sites or content by relying on the similarities of words. It is applied in various contexts, including fake domains [7], [15], phishing campaigns [18], [19], and the hijacking of smart speakers [11], [27]. Several cyber-squatting strategies are hypothesized and demonstrated in practice, including simple/frequent typos [1], [8], [20], [25], the visual similarity of words [7], and collocation of common words [10], [14].

Sound-squatting is a phishing technique that leverages words with pronunciations similar to legitimate domain names to trick users. It receives relatively little attention compared to other types of cyber-squatting and is potentially gaining traction with the advent of smart speakers, voice assistants [11], and the resurgence of content consumption exclusively via audio, such as radio programs and podcasts where some website is advertised. Detecting this type of squatting is challenging due to variations in pronunciation across languages and individuals [26]. Previous studies focus only on lists of English homophones [15], thus lacking coverage in terms of non-existing words, words with similar pronunciation (quasi-homophones), and cross-language scenarios.

Several recent works longitudinally evaluate the occurrence of domain squatting using tools to create candidates and verify the existence of the created domains. Even further, some works evaluate the existence of Transport Layer Security (TLS) certificates for supporting phishing, certificates issued by actual Certificates Authorities (CA) [5], [9], [17], [23]. However, these works focus only on techniques such as typo-squatting, combo-squatting, and homograph-squatting, ignoring sound-squatting.

We introduce Sound-squatter, an AI-based system that is capable of *automatically* creating sound-squatting candidates. Sound-squatter generates candidates for any given target name, works at the sub-word level, and allows configurable approximations during the search for candidates. It naturally supports multiple languages and the cross-language scenario (described next). Sound-squatter is built using a state-of-the-art Transformer Neural Network [24] trained to produce candidates in any language. It receives both the written form of the word (*grapheme*), its pronunciation, represented using the International Phonetic Alphabet (IPA), and the language the word belongs to. At inference, Sound-squatter uses its sequence-to-sequence model to find written alternatives with similar pronunciations in the target language.

Important, any sound-squatting protection mechanism needs to consider multi-language scenarios. However, this is not a trivial task since each language, accent, and speaker's proficiency level impacts the way people read and understand words. Sound-squatter is a multi-language model that can generate cross-language candidates, considering foreign pronunciations that are similar to the given word.

We first validate Sound-squatter by comparing the list it generates against known homophones. Here we find that Sound-squatter can automatically generate around 80% of the known homophones for any specific language and thousands of additional quasi-homophones, including cross-language homophones if specified. Next, we collect domain names as appeared in the past 24 days in the TLS certificates using the Certificate Transparency (CT) logs [12]. We then check whether sound-squatting candidates for top-popular websites appear in the feed.

In summary, our contributions are as follows:

- **A methodology for a multi-language sound-squatting generation.** We present a methodology for generating multi-language and cross-language sound-squatting candidates using contextual phoneme-based generation. This approach can be particularly useful for enhancing the security of less central countries and correlated spoken languages.

DOI 10.1109/EuroSPW59978.2023.00063

- **A broad study of the abuse of sound-squatting in domain registration using CT logs.** We conduct a comprehensive study of sound-squatting abuse in domain registration by analyzing candidates generated by Sound-squatter using a set of 200 million certificates collected on CT logs. Our findings show that approximately 6.5% of the generated candidates are registered, and around 8.2% of the target domains may be subject to abuse.
- **Comparison of sound-squatting and other squatting techniques.** We compare the search results of sound-squatting with those of other squatting techniques generated with state-of-the-art algorithms. Our findings demonstrate that sound-squatting poses a threat with different behavior than, e.g. typo-squatting, but its potential abuses are comparable to those of other squatting techniques.

The use of Sound-squatter can provide protection against attacks that target names such as brands. It offers an affordable solution for searching squatting campaigns in less controlled markets by automatically generating potential domain names that attackers can abuse or have already abused. Additionally, Sound-squatter can be used to compile search lists to monitor certificate issuance or domain registration, thus enhancing protection against zero-day phishing campaigns.

In Section 2, we describe Sound-squatter, and in Section 3, we validate it. Then, in Section 4.1, we discuss the results of domain name sound-squatting, and in Section 4.2, we show our findings. We also discuss related work in Section 5 before concluding the paper in Section 6. In Appendix A, we provide background information.

2. Sound-squatter: System Description

Sound-squatter is an AI-based system capable of creating sound-squatting candidates that relies on Transformer Neural Network to translate from phonemes to graphemes. The model operates as depicted in Figure 1. The generation pipeline consists of four main components:

1) The ***Grapheme to Phoneme*** (G2P) component transforms an input word written in grapheme form (such as the word `eye`) into its corresponding International Phonetic Alphabet (IPA) representation (such as `/aɪ/` for British English).
2) The ***IPA Encoder*** component encodes the IPA word into a vector latent representation.
3) The ***Grapheme Decoder*** (P2G) component interactively decodes the vector representation into characters to compose graphemes form that are quasi-homophones of the input word (such as `I`) in a target language.
4) The ***Post Processor*** component performs beam search over the logits and selects tokens based on the probability from the decoder's output, resulting in the generation of multiple quasi-homophones from a single pronunciation.

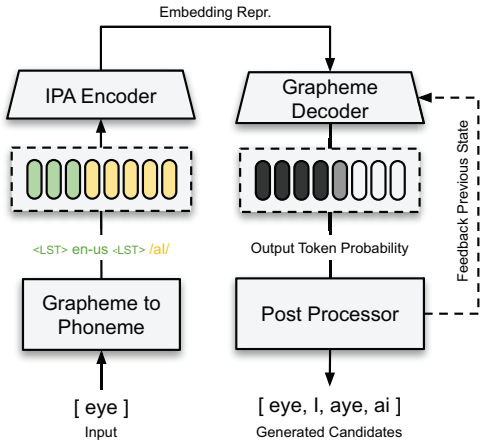


Figure 1: Architecture used during inference. The process to generate candidates comprises an *IPA Encoder* that maps the input to a latent space and a *Grapheme Decoder* that several ways to reconstruct the input.

2.1. *Grapheme to Phoneme*

The *Grapheme to Phoneme* module is quite standard and different options exist to derive the IPA representation of a word grapheme, many of which that include multi-language solutions. The most common ones are rule-based and data-driven models. Rule-based G2P models use hand-crafted rules to transliterate the pronunciation of a word based on its orthographic features. These models are relatively simple but require expert knowledge and linguistic resources to be developed. Data-driven models usually rely on machine learning and annotated data.

Sound-squatter can use any G2P model as long it keeps consistency in the symbol usage. We chose to use eSpeak NG (Next Generation) text-to-speech engine[1] and Epitran[2] for transliterating orthographic text into IPA. We chose two these models because, while eSpeak NG is a better fit for our application in English-GB and English-US, Epitran is more suitable for other languages, because eSpeak NG often switch languages when transliterating known words, which breaks the core idea of this proposal.

2.2. *IPA Encoder* and *Grapheme Decoder*

Sound-squatter key components are the *IPA Encoder* and *Grapheme Decoder*. We use state-of-the-art Transformer Neural Networks [24]. Transformer rely on the self-attention mechanism to learn how to translate a word in IPA back into grapheme format. The output of *IPA Encoder* is forwarded to *Grapheme Decoder*, which uses the contextual information extracted from the input by *IPA Encoder*, together with the current state of the output. The Transformer is an auto-regressive model at inference (See A.3), i.e., it can leverage its own past history of predictions to forecast future states.

In particular, given an IPA input and the predictions made for each of the previous $(N-1)$ characters,

1. https://github.com/espeak-ng/espeak-ng (accessed on 10/10/2022).
2. https://github.com/dmort27/epitran (accessed on 10/10/2022).

Grapheme Decoder forecasts the probabilities of each possible character being the N-th character. It is then the role of *Post Processor* to look at these probabilities and feed the history back to the *Grapheme Decoder* for the new forecast.

For training we rely on labelled data and follow standard algorithms used to train Transformer Neural Networks for causal language models. We design our solution as a single multi-language model, with the explicit capability of generating cross-language homophones and quasi-homophones. We control the language used to read the grapheme by changing the language or accent of the G2P model. For example, the word "water" has different pronunciations in English-US (/ˈwɑtəɹ/), English-GB (/ˈwɔːtə/). In the cross-language case, for example, suppose we pronounce "water" in English-US and specify that we want the grapheme form to be transliterated into French-FR. In that case, the model can generate the quasi-homophone "warères" (/waʁəʁ/), which does not exist as a word in French-FR but has a similar pronunciation to "water" in English-US.

To specify the language the transformer shall use to transliterate the phoneme back into grapheme form, we add a language context to the input by introducing a special token called the "Language Special Token" (LST) to provide contextualized information in the input. In a nutshell, during training, the LST provides the input and output language information to the transformer, which learns to transliterate the phoneme based on the information contained between them.

2.3. *Post Processor*: the Quasi-Homophone Generation

The *Post Processor* is the last element of the inference: it receives as input the *Grapheme Decoder*'s probabilistic forecasts of the next character, and it keeps track of the history of predictions to feedback to *Grapheme Decoder*. Because *Grapheme Decoder* operates as an autoregressive model at inference, we can generate more than a candidate quasi-homophone by tweaking the history the *Post Processor* feeds back. We use a Beam Search to pick from amongst the tokens those whose probabilities add up to p, also known as `top-p` strategy.

Figure 2 shows the exact output for four iterations. At each step, the *Post Processor* stores C most-likely predictions of *Grapheme Decoder* whose probabilities add up to at least p and constructs alternative histories for the next step. Figure 2 shows this process with a directed graph diagram (with $p = 0.8$) starting from the IPA representation of `eye`. After four iterations, the process generates six ways to write `eye`. Each branch stops when *Grapheme Decoder* outputs the special character `EoS`.

The number of iterations (M), maximum number of candidates predictions (K) and probability (p) are parameters that we can define manually. For this work, we empirically define the parameters as follows: *Post Processor* iterates $M = N + 6$ times, where N is the size of the source string, and we limit exploration with $K = 100$ and $p = 0.8$, where K is the maximum number of possible candidates generated. These parameters have been determined by manual inspection and depended on the specific task and dataset.

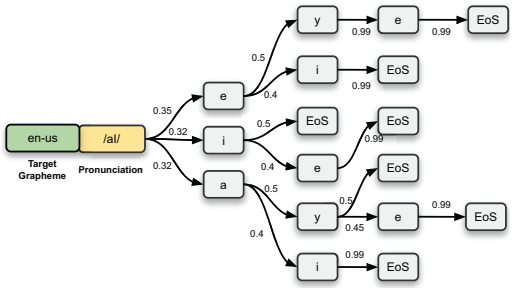


Figure 2: Illustration of the inference process with $p = 0.8$. At each inference step, we explore n next characters whose probabilities add up to p. For readability, we round probabilities in the figure.

TABLE 1: Dataset size for each chosen language.

Language Tag	Language	Region	Data size
en-GB	English	United Kingdom	65118
en-US	English	United States	125923
fr-FR	French	France	245971
	Total		437012

3. Sound-squatter: Training and Validation

In this section, we describe the training process and the validation for our model. We selected three different forms of pronunciation: English-US, English-GB, and French-FR. We chose English because it is the most widely used language on the Internet, and two English variations to illustrate this factor is important in homophone-based impersonation. We also chose French-FR because it is also a not a phonetic language, which is more prone to confusion during transliteration.

Phonetic languages have a close correspondence between pronunciation and written representation, with each letter or character representing a specific sound. While French and English have some phonetic elements, they are less purely phonetic than languages with more straightforward sound-to-spelling correspondences, such as Italian and Spanish.

3.1. Dataset and Training

The training dataset consists of the list of English-US, English-GB, and French-FR words from the GNU Aspell [3] word list. GNU Aspell is a free and open-source spell checker containing word lists for multiple languages. To acquire the pronunciation, we use rule-based G2P tools: eSpeak NG for English-GB and English-US and, Epitran for French-FR.

Table 1 shows the size of the dataset for each language. In total, we use 437 012 words and their pronunciation.

We train the *IPA Encoder* and the *Grapheme Decoder* with a batch size of 256 words. The training set contains 80% of the samples, and we preserve 10% for validation and 10% for test sets. The maximum sequence length is 50 tokens. We use the validation set to select the best model and the test set to verify over-fitting. We use the Adam optimizer with $LR = 0.0001$, $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 10^{-9}$. We train the model for 10 epochs, which takes

around 10 minutes on a single NVIDIA Tesla v100. The Transformer Network hyperparameters for the encoder and the decoder are symmetric and are defined as follows: the size of the hidden representation is 512, the embedding dimension is 512, the number of heads is 8 and, the vocabulary size is 123.

3.2. Model Validation

We verify whether the model generates homophones by comparing its results against known sets of homophones. We consider as homophones any group of words having the same IPA sequence. Then, we select one word for each set and ask Sound-squatter to generate candidates. The expected result is a considerable intersection between known homophones and generated candidates.

We randomly select from our validation data 2 000 pronunciations with more then one written form for each language. We generate 42 716 candidates, i.e. on average, we produce 7.11 candidates per target. Sound-squatter finds 77.86% of the known homophones, thus showing that Sound-squatter can generate words with the same exact pronunciation but different spelling. In addition, Sound-squatter generates many possible alternate spellings of the same target words. The remaining words (28 012) are not necessarily existing dictionary words but show similarities with the target word. We call them quasi-homophones.

4. Domain Sound-squatting in the Wild

4.1. Search Methodology

Our approach to identifying the eventual presence of cross-language sound-squatting domains involves a systematic process that begins with generating candidates using Sound-squatter. This allows us to create a large number of possible domains quickly and efficiently.

We select three languages: English-US, English-GB, and French-FR. We generate the same and cross-language sound-squatting for all possible combinations, resulting in 9 combinations in total. As a baseline, we use the AIL typo-squatting tool [16] to generate other-squatting candidates for a given domain. Although it AIL tool also includes homophone generation, we find that it generates only 4 sound-squatting candidates in total, a negligible amount compared to what Sound-squatter does.

As a starting point for identifying popular websites likely to be targeted by squatting attacks, we consider the list of top-accessed websites from SimilarWeb, a leading website analytics provider. Next, we use the two generative tools to generate candidates. Here, we stick to the second-level domain due to the nature of the domain impersonation we want to focus on.

To identify potential squatting domains, we search for our candidate homophones on TLS certificates of registered domains using CertStream [3] (Certificate Transparency Logs). TLS certificates are used to secure websites and provide information about the domain owner's identity. We extract all the server names to build a list of all registered second-level domains to check against our candidates. This is the most time-consuming part due to the overwhelming amount of data. By searching on these names, we can identify domains that potentially could impersonate popular websites very early in their life-cycle. This step is crucial in identifying squatting attacks.

3. https://certstream.calidog.io/

TABLE 2: Generated candidates for cross-language sound-squatting by language combination and for typo-squatting.

Sound-squatting				Other-squatting
	Language G2P			
Language P2G	en-GB	en-US	fr-FR	
en-GB	15 427	14 210	23 910	
en-US	16 526	14 878	24 984	
fr-FR	19 218	15 852	14 103	
Overall	140 581			360 652

4.2. Analysis of Registered Candidates

During 24 days from February 8 to March 3, 2023, we collect a total of 208 652 973 certificate entries. These certificates contain 44 876 072 unique registered domains, of which 38 083 322 are unique 2nd-level domain names [4]. These certificates were issued by 202 organizations located in 46 different countries and across 4 037 Top Level Domains (TLDs).

4.2.1. Quantitative analysis. From SimilarWeb[5] we select the top 1 000 most accessed domains.[6] Using Sound-squatter we generate 159, 108 unique names from the 1, 000 candidates in the same 9 cross-language configuration. Table 2 details the number of candidates by combination. Note that the amount of candidates is quite regular for the same-language homophone generation. Still, when the G2P is configured for fr-FR, there is a significant increase in the number of generated candidates. Since fr-FR is a complex, not phonetic language, the association with en-US and en-GB gives some relevant possibilities for the P2G module.

For comparison, the AIL typo-squatting tool generates 360 652 unique candidates from the same 1 000 domains, using 14 different algorithms that modify the second-level domain, such as omission, repetition, replacement, double replacement, keyboard insertion, addition, strip dash, vowel swap, add a dash, bit-squatting, homoglyph, common misspelling, homophones, and singular pluralize.

Upon searching for these candidates in the stream of certificates, we find 5 946 unique sound-squatting candidate names that cover 820 target domains potentially being abused by over 13 495 registered domains. This divergence between the number of names and domains can be attributed to our search being limited to the second-level domain, which considers domains with different TLDs as a single name. For instance, for `gugle.com`

4. CT logs includes both new certificates and updates to existing certificates

5. Accessed at 19/09/2022 using a free account

6. We collect the most popular domains that do not contain numbers and acronyms or those that have a high likelihood of being randomly generated because they are less likely to be used for sound-squatting attacks. We filter out 174 to reach 1000 domains

TABLE 3: The percentage of sound-squatting candidates generated for each language combination and other-squatting that were found to exist in at least one domain registered.

Sound-squatting				Other-squatting
	Language G2P			
Language P2G	en-GB	en-US	fr-FR	
en-GB	10.26%	10.40%	4.63%	
en-US	9.20%	9.60%	4.48%	
fr-FR	10.84%	11.62%	9.40%	
Overall		6.25%		8.98%

TABLE 4: Total of targets potentially abused by sound and other-squatting. Notice that some domains are present in a single combination.

Sound-squatting				Other-squatting
	Language G2P			
Language P2G	en-GB	en-US	fr-FR	
en-GB	520	503	444	
en-US	497	478	450	
fr-FR	570	556	532	
Overall		820		959

and `gugle.tv` being two different domains targeting the same `google` domain. Table 3 shows the percentage of candidates registered. Interestingly, the ratio for the same language is similar across all languages.

Table 4 details the number of candidate domains that we find to be potentially abused. Note that some domains are potentially abused in a unique cross-language combination since the global unique abused domains are 820. This provides strong evidence that sound-squatting also considers cross-language homophones. Considering other-squatting, we find 32 408 names potentially being abused for 51 371 unique registered domains, which represents a total of 8.98% of the generated candidates, for completeness, 30 586 names were exclusively found by AIL. In total, 959 domains are potentially squatted by at least one candidate using the typo-squatting technique. All in all, the number of candidates found by sound- and other-squatting domains are similar.

In summary, the Venn Diagram of names shows the set of candidates generated by Sound-squatter, the set of candidates generated by AIL tool and all the names present in the collected certificates. It clearly shows that the cross-language sound-squatting has a small intersection with the set of other-squatting with 4 124 domains being found exclusively by Sound-squatter.

4.2.2. Quality of quasi-homophones. We finally assess the quality of the set of homophones and quasi-homophones generated by Sound-squatter. Soundex [21] and Levenshtein Normalized Index of Similarity (or Levenshtein ratio) [13] are normally used to evaluate the quality of homophones. The Soundex algorithm is used to calculate valid phonetic representation. Specifically, we compare targets with candidates using Soundex. We consider bad those with low quality scores. The Levenshtein ratio measures candidates with the original target's spelling. It returns a value in between 0 (dissimilar) and 1 (identical). We consider bad those candidates with a Levenshtein ratio distance smaller than 0.5. We find 140 581 sound-squatting candidates to pass both filters, i.e., only around 11% are of bad quality.

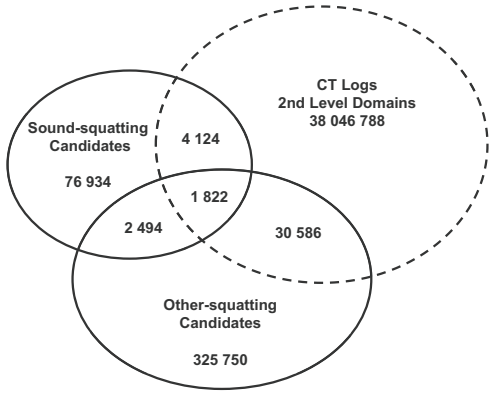
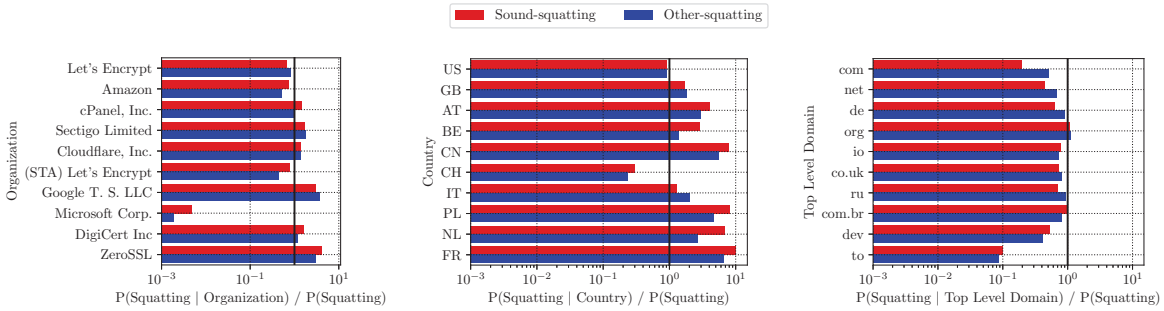


Figure 3: Venn Diagram of the generated names comparing the sound-squatting and other-squatting technique. The dashed border for CT Logs names indicates that we are only comparing with our 24 days of observation.

TABLE 5: Top 20 in SimilarWeb Rank and the registered squatting domains found for each. We consider the TLD.

	Squatting Type		
Target Domain	Other	Sound	Intersection
google	236	69	10
youtube	145	103	4
facebook	97	23	4
instagram	82	2	1
twitter	78	34	1
baidu	121	146	7
yandex	56	19	4
xvideos	172	15	0
xnxx	163	15	7
wikipedia	71	22	5
pornhub	112	41	3
amazon	317	79	6
yahoo	108	57	6
tiktok	115	78	2
live	344	26	4
linkedin	74	7	1
reddit	80	99	8
whatsapp	144	46	6
xhamster	212	24	0
netflix	108	28	6

4.2.3. Comparative behaviour in popular domains. In Table 5, we compare the most popular domains in the SimilarWeb Rank and the number of registered domains found in the dataset by technique. Notice again the commonly found names are very few. The behaviour for sound-squatting is quite different from that for typo-squatting, for instance. Sound-squatting tends to be more successful in less known domains, where the written form is less certain. Other-squatting instead targets the most popular brands to profit from user mistakes and not brand unfamiliarity. This is particularly worrisome, for instance, in a scenario where a domain might be said and the users need to phonetically transcribe it, e.g., in audio advertisements.



(a) (b) (c)

Figure 4: We compute the $P(Squatting \mid X)/P(Squatting)$. Where X is Organization, Country or TLD. A divergence in the ratio indicates some bias in the attackers choice for squatting.

4.2.4. Issuer and TLD characterization. The issuer field on a certificate contains important information such as the organization's name and other identifying data. Using this data, we compute the $P(Squatting \mid X)/P(Squatting)$, with X being either i) the subsets of Top-10 certificate organizations, ii) issuer's country, or iii) TLD of the registered domain (Figure 4). Any ratio different from 1 (it is indicated by a vertical line) is an indication of a correlation between the variables, i.e., less (< 1) or more (> 1) probable of hosting abused domains.

In Figure 4a, we notice that ratio for `ZeroSSL` and `Google Trust` is way above 1 suggesting high correlation between these organization and squatting. Conversely, Microsoft shows an extremely low probability of issuing a squatting-related certificate.

Figure 4b also suggests that the probability of having a squatting attack given a certificate issued in France is 10 times the average squatting rate in the dataset. Here it is worth mentioning that we train Sound-squatter also on French words so we expect more possible victims in FR CAs. The TLD data on Figure 4c indicates, instead, that the most popular TLDs do not seem to be particularly targeted for squatting as the probability ratio is at most 1. A possible reason for this is linked to the costs and availability in the registration of new domains with these TLDs.

5. Related Work

Nikiforakis et al. discovered sound-squatting as a phishing technique by generating candidates from a static database of English homophones [15]. They replaced words in domains with homophones, evaluate, and categorize the candidates. Sonowal and Kuppusamy focus on phoneme-based squatting for visually impaired people [18], [19]. They propose systems to detect phoneme-based phishing in the accessibility interface and evaluate exposure to phishing in email campaigns, proposing a manual technique to detect phishing emails.

In [9], the authors evaluate how CAs are involved in the HTTPS phishing ecosystem. In particular, what insecure practices of CAs can lead to the increase of the attacks. It show the high risk that squatting domains present in a TLS environment due to end-users, who may have just glanced at the browser's address bar, can believe the squatting domains to be legitimate. The authors focus their report on combos, typo, and homograph squatting. While [17] follows a similar methodology to uncover the target embedding squatting technique that embeds an entire target domain, unmodified, using one or more subdomains of the actual domain. Using all HTTPS certificates they perform a longitudinal analysis of how target-embedding impersonation has evolved.

While [17] describes a methodology to uncover target embedding squatting. The authors perform a longitudinal analysis of how this technique has evolved using all HTTPS certificates.

6. Conclusion

We introduced Sound-squatter, an AI-powered cross-language sound-squatting generator. We used Transformer trained to model both the phoneme representation and the pronunciation of words, producing high-quality homophones, and not-existing words with similar pronunciations. We added context to the input to control the target languages during inference via Language Special Token.

We validated Sound-squatter's capability to find known homophones in English-US, English-GB, and French-FR, explored the possibility of cross-language sound-squatting, and compared it with other-squatting techniques. We selected the most popular domains and generated sound-squatting and typo-squatting candidates for these domains. Upon searching for these candidates in the stream of certificates, we found that sound-squatting is not covered by traditional squatting techniques (intersection between sound and other squatting represents 2.3% of Sound-squatter candidates). Yet, sound-squatting is as potentially popular as other-squatting mechanisms, but it tends to target less-know domains than other-squatting. We also extracted metrics from certificates that may have contained impersonating domains and presented the results as a preliminary study of a possible sound-squatting abuse investigation.

For future work, we plan to investigate the use of Sound-squatter for the protection of smart speakers and software packages' names. We also plan to investigate multi-modal models to include sound features directly in the model training.

References

[1] Pieter Agten, Wouter Joosen, Frank Piessens, and Nick Nikiforakis. Seven months' worth of mistakes: A longitudinal study of typosquatting abuse. In *Proceedings of the 22nd Network and Distributed System Security Symposium (NDSS 2015)*. Internet Society, 2015.

[2] International Phonetic Association. *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press, 1999.

[3] K. Atkinson. Gnu aspell 0.60.4. GNU Aspell). Retrieved from http://aspell.net, 2006.

[4] MITRE ATT&CK. Capec-616: Establish rogue location. https://capec.mitre.org/data/definitions/616.html, 2022.

[5] Zakir Durumeric, James Kasten, Michael Bailey, and J. Alex Halderman. Analysis of the HTTPS certificate ecosystem. In *Proceedings of the 2013 conference on Internet measurement conference*. ACM, October 2013.

[6] Dyslexia Reading Well. The 44 Phonemes in English. Online, 2022. Accessed: October 5, 2022.

[7] Tobias Holgers, David E Watson, and Steven D Gribble. Cutting through the confusion: A measurement study of homograph attacks. In *USENIX Annual Technical Conference, General Track*, pages 261–266, 2006.

[8] Mohammad Taha Khan, Xiang Huo, Zhou Li, and Chris Kanich. Every second counts: Quantifying the negative externalities of cybercrime via typosquatting. In *2015 IEEE Symposium on Security and Privacy*, pages 135–150. IEEE, 2015.

[9] Doowon Kim, Haehyun Cho, Yonghwi Kwon, Adam Doupé, Sooel Son, Gail-Joon Ahn, and Tudor Dumitras. Security analysis on practices of certificate authorities in the HTTPS phishing ecosystem. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*. ACM, May 2021.

[10] Panagiotis Kintis, Najmeh Miramirkhani, Charles Lever, Yizheng Chen, Rosa Romero-Gómez, Nikolaos Pitropakis, Nick Nikiforakis, and Manos Antonakakis. Hiding in plain sight: A longitudinal study of combosquatting abuse. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 569–586, 2017.

[11] Deepak Kumar, Riccardo Paccagnella, Paul Murley, Eric Hennenfent, Joshua Mason, Adam Bates, and Michael Bailey. Skill squatting attacks on amazon alexa. In *27th USENIX security symposium (USENIX Security 18)*, pages 33–47, 2018.

[12] Ben Laurie. Certificate transparency. *Communications of the ACM*, 57(10):40–46, 2014.

[13] Vladimir Iosifovich Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966. Doklady Akademii Nauk SSSR, V163 No4 845-848 1965.

[14] Pablo Loyola, Kugamoorthy Gajananan, Hirokuni Kitahara, Yuji Watanabe, and Fumiko Satoh. Automating domain squatting detection using representation learning. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 1021–1030, 2020.

[15] Nick Nikiforakis, Marco Balduzzi, Lieven Desmet, Frank Piessens, and Wouter Joosen. Soundsquatting: Uncovering the use of homophones in domain squatting. In *International Conference on Information Security*, pages 291–308. Springer, 2014.

[16] AIL Project. Ail-typo-squatting. https://ail-project.github.io/ail-typo-squatting/, Accessed: 2023.

[17] Richard Roberts, Yaelle Goldschlag, Rachel Walter, Taejoong Chung, Alan Mislove, and Dave Levin. You are who you appear to be. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. ACM, November 2019.

[18] G Sonowal and KS Kuppusamy. Mmsphid: A phoneme based phishing verification model for persons with visual impairments. information and computer security journal, 2019.

[19] Gunikhan Sonowal. A model for detecting sounds-alike phishing email contents for persons with visual impairments. In *2020 Sixth International Conference on e-Learning (econf)*, pages 17–21. IEEE, 2020.

[20] Janos Szurdi, Balazs Kocso, Gabor Cseh, Jonathan Spring, Mark Felegyhazi, and Chris Kanich. The long {"Taile"} of typosquatting domain names. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 191–206, 2014.

[21] Thottingal, Santhosh. IndicSoundex Algorithm. Online, 2009. Accessed: 2023-03-22.

[22] Ke Tian, Steve T. K. Jan, Hang Hu, Danfeng Yao, and Gang Wang. Needle in a haystack: Tracking down elite phishing domains in the wild. In *Proceedings of the Internet Measurement Conference 2018*, IMC '18, page 429–442, New York, NY, USA, 2018. Association for Computing Machinery.

[23] Ivan Torroledo, Luis David Camacho, and Alejandro Correa Bahnsen. Hunting malicious TLS certificates with deep neural networks. In *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security*. ACM, January 2018.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[25] Yi-Min Wang, Doug Beck, Jeffrey Wang, Chad Verbowski, and Brad Daniels. Strider typo-patrol: Discovery and analysis of systematic typo-squatting. *SRUTI*, 6(31-36):2–2, 2006.

[26] Yuwei Zeng, Tianning Zang, Yongzheng Zhang, Xunxun Chen, and YiPeng Wang. A comprehensive measurement study of domain-squatting abuse. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pages 1–6, 2019.

[27] Nan Zhang, Xianghang Mi, Xuan Feng, XiaoFeng Wang, Yuan Tian, and Feng Qian. Dangerous skills: Understanding and mitigating security risks of voice-controlled third-party functions on virtual personal assistant systems. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 1381–1396. IEEE, 2019.

TABLE 6: Examples of phonemes for American English with the grapheme and some examples (source [6]).

Type	Phoneme	Grapheme
Consonant	ʃ	sham, ocean, sure, special, pension
Consonant	v	vine, of, stephen, five
Vowel	æ	cat, plaid, laugh
Vowel	eɪ	bay, maid, weigh, straight, pay

A. Background

A.1. Cyber-squatting and sound-squatting

Cyber-squatting is a type of attack where malicious actors impersonate legitimate resources [4], often through domain-squatting, which involves registering fake domain names to divert traffic from popular websites. In 2018, over 657 k domain names impersonating 702 popular brands [22] were identified through an in-depth search of more than 224 million DNS records [22]. Sound-squatting, a form of domain-impersonation, exploits the assumption that targets may confuse words with similar pronunciations and can occur in various contexts, such as with Alexa voice assistant abuses. Mitigating sound-squatting requires extra DNS checks and purchasing potential squatted domains, but to do so effectively, targeted brands need to know which names attackers might use to impersonate them. Sound-squatter helps generate ranked candidate names to proactively mitigate the problem.

A.2. International Phoneme Alphabet (IPA)

The International Phonetic Alphabet represents phonemes with standard symbols (IPA) [2]. Table 6

shows that some phonemes may be equivalent to more than one grapheme. Some languages are phonetically consistent, like Italian and German, where most graphemes correspond to a single phoneme. Other languages, like English, are not phonetically consistent, leading to pronunciation confusion, and are more susceptible to sound-squatting attacks.

Sound-squatter uses IPA to encode the input words. There exist solutions that translate any word in the corresponding IPA. For instance the eSpeak NG (Next Generation) Text-to-Speech engine supports more than 100 languages. Sound-squatter uses it to derive the input IPA given a target word and language.

A.3. Transformers Neural Networks

Transformers are a reality in the deep neural network community. Initially proposed for translation [24], they have been used as the means to achieve natural language processing (NLP), computer vision (CV), and speech processing. Transformers consist of an encoder and decoder, each with N identical blocks, including MultiHead Attention and feed-forward layers. The decoder adds a cross-multihead attention mechanism that computes the attention between input and previous states of the target. During training, it uses a mask to learn multiple states. In inference, the decoder receives the previous states generated to compute cross-attention, allowing correct prediction of the next character in a word or the next word in a sentence.

Sound-squatter uses Transformers to generate words at the character level and using a beam search to generate more than one single candidate. In consequence, Sound-squatter generates alternative ways to write the same input IPA sequence.