What a SHAME: Smart Assistant Voice Command Fingerprinting Utilizing Deep Learning

Jack Hyland
jxh3105@rit.edu
Rochester Institute of Technology

Conrad Schneggenburger
ccs5486@rit.edu
Rochester Institute of Technology

Nick Lim
njl3087@rit.edu
Rochester Institute of Technology

Jake Ruud
jlr6304@rit.edu
Rochester Institute of Technology

Nate Mathews
nate.mathews@mail.rit.edu
Rochester Institute of Technology

Matthew Wright
matthew.wright@rit.edu
Rochester Institute of Technology

ABSTRACT

It is estimated that by the year 2024, the total number of systems equipped with voice assistant software will exceed 8.4 billion devices globally. While these devices provide convenience to consumers, they suffer from a myriad of security issues. This paper highlights the serious privacy threats exposed by information leakage in a smart assistant's encrypted network traffic metadata. To investigate this issue, we have collected a new dataset composed of dynamic and static commands posed to an Amazon Echo Dot using data collection and cleaning scripts we developed.

Furthermore, we propose the Smart Home Assistant Malicious Ensemble model (SHAME) as the new state-of-the-art Voice Command Fingerprinting classifier. When evaluated against several datasets, our attack correctly classifies encrypted voice commands with up to 99.81% accuracy on Google Home traffic and 95.2% accuracy on Amazon Echo Dot traffic. These findings show that security measures must be taken to stop internet service providers, nation-states, and network eavesdroppers from monitoring our intimate conversations.

CCS CONCEPTS

• **Security and privacy**;

KEYWORDS

smart assistant, voice command, traffic fingerprinting, deep learning

ACM Reference Format:
Jack Hyland, Conrad Schneggenburger, Nick Lim, Jake Ruud, Nate Mathews, and Matthew Wright. 2021. What a SHAME: Smart Assistant Voice Command Fingerprinting Utilizing Deep Learning. In *Proceedings of the 20th Workshop on Privacy in the Electronic Society (WPES '21), November 15, 2021, Virtual Event, Republic of Korea.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3463676.3485615

WPES '21, November 15, 2021, Virtual Event, Republic of Korea

ACM ISBN 978-1-4503-8527-5/21/11…$15.00
https://doi.org/10.1145/3463676.3485615

1 INTRODUCTION

Smart home assistants, such as the Google Home and Amazon Echo, are becoming increasingly common in modern households [29]. Thanks to their convenience, low cost, and wide range of functions, these devices continue to expand their role in performing everyday tasks. With the growing purchase and use of these smart assistants worldwide comes questions concerning end-user privacy. In this work, we investigate the extent to which smart assistants leak information through a type of attack called Voice Command Fingerprinting (VCF). In this attack, an adversary eavesdrops on a smart speaker's encrypted traffic and attempts to infer the voice command spoken by the victim. Since most data in transit is encrypted via Transport Layer Security (TLS), this attack utilizes side-channel information such as packet size, direction, and timestamps to infer the voice command based on patterns, bypassing the need for decryption. By analyzing this information, a mapping of traffic flow patterns to voice commands can be created.

Deep learning (DL) models have proven effective in fingerprinting attacks, such as Website Fingerprinting (WF) [23] and VCF [30]. However, supervised deep learning models generally rely on large-scale datasets for training. Previous works in VCF have collected their data manually [13], resulting in small datasets that would not be sufficient for training DL models. Wang et al., in the most recent work related to VCF [30], were able to automate the data collection process, resulting in a large-scale dataset. This paper proposes a more streamlined approach to automate the process while also reliably removing erroneous samples with minimal human interaction.

Most of the recent work in WF is focused on Tor, which has fixed 512-byte *cells* that greatly limit the value of packet size information [19, 23]. VCF data, however, typically runs over TLS connections that reveal fine-grained packet sizes to an eavesdropper. To account for this, we design a new ensemble model, SHAME, that captures packet direction, time, and packet size.

The contributions of this work are:

(1) We design and develop an automated data collection system for smart assistants to generate large-scale datasets. We subsequently use this system to collect an open-source dataset of 10 dynamic, and 10 static voice commands for Amazon Alexa devices [11].
(2) We develop a novel VCF model, SHAME, that uses multiple base models to cover timing, inter-packet delay, and packet size separately before putting them into an ensemble. We

show in experiments with our dataset that SHAME achieves state-of-the-art VCF performance.

The code for our automated data collection script[1], SHAME model[2], and proof of concept[3] attack video [10] can all be referenced on Github.

2 RELATED WORK

Smart Home Assistants. Knowing the basic operation of smart home assistants is important to understand VCF. The device listens for a *"wake word"* that starts an audio recording of the user's command—once heard, the assistant records until ambient audio levels are detected. Finally, the entire recording, including the wake word, is sent to the cloud for processing. Using a combination of Automatic Speech Recognition (ASR) and Natural Language Processing (NLP), the speech files are transcribed into text. That text is then sent to an API, and an appropriate response is determined. Finally, the response is sent back as text to the device, converted into audio using Text-to-Speech software, and spoken back to the user. A more detailed explanation of the system and its inner workings is outlined by Li et al. [15]. Recently, many works have investigated vulnerabilities of these devices, primarily targeting weaknesses of the ML systems that power voice-activated smart systems [5, 21, 28, 32, 33]. These attacks craft malicious commands that trick the device into performing some erroneous action.

Traffic Fingerprinting. The privacy threats posed by the information leakage of traffic metadata have been considered since long before smart devices [14]. One particularly popular attack to explore has been website fingerprinting (WF). WF attacks deanonymize private Internet browsing by associating websites with the traffic metadata they produce. Recently, many ML and DL techniques have proven effective against Tor with and without defenses implemented [4, 9, 18, 20, 23, 24]. Additionally, several works have also examined the fingerprintability of IoT devices [2, 3, 6, 13, 25]. These works have most notably shown that it is possible to uniquely identify smart home assistants, such as Amazon Alexa and Google Home devices, solely using the network traffic they produce.

3 AUTOMATED VCF DATA COLLECTION

3.1 Overview

We aim to demonstrate the possibility for an attacker to automate the collection, processing, and training of smart assistant network traffic. Automating this process requires collecting a dataset of voice commands against a smart assistant, writing software to prompt the device, collecting network captures associated with the response, and developing a fingerprinting model to analyze the collected traffic to identify features. We also aim for a modular design to enable other researchers to collect traffic from multiple device types.

We established a set of twenty questions to measure the effect that dynamic questions have on deep-learning VCF models, shown in Appendix [C]. Here, we have ten *static* questions, while the remaining ten questions are classified as *dynamic*. Dynamic questions are selected based on whether the elicited response is likely to change for each query iteration, whereas static questions produce consistent responses regardless of when the question is queried.

3.2 Data Collection

The data collection framework is centered around a Raspberry Pi 4 running Raspberry Pi OS (Linux 5.10.17-v7l+). The Raspberry Pi is equipped with one Ethernet port and one wireless antenna. Running *hostapd* [16] allows the Pi to act as a bridged access point for the smart device to reach the wide-area network. This arrangement allows for complete visibility of Echo Dot's wireless network traffic without using port mirroring on dedicated networking equipment.

We selected the Amazon Echo Dot (2nd Generation) as our primary testing device as it includes a 3.5mm audio jack capable of auxiliary sound output. By connecting the output jack of the Echo Dot directly to a modified input jack of the Raspberry Pi, we can dynamically detect the moment when the response from the voice assistant finishes without any concern of external noise interfering (see Figure [1]).

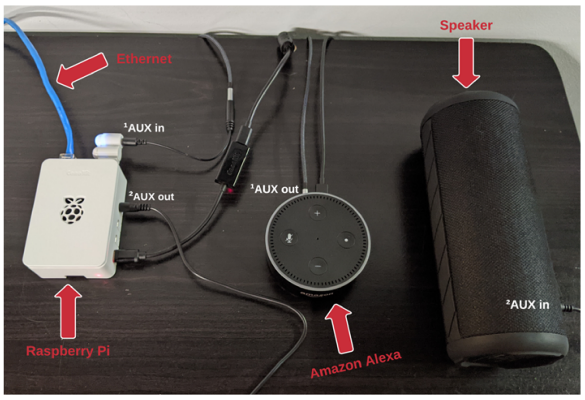

Figure 1: Our automated voice traffic collection setup.

We use synthetic voices generated by *festival-lite* [8] creating a variety of audio command files consisting of several distinct voice models for each question. Furthermore, to maximize the diversity of available input samples, the script also generates several additional audio variants by altering the original voice models' pitch, speed, and intonation.

3.3 Data Cleaning

When monitoring our collection process, we noticed frequent erroneous captures. We define a *bad capture* as any capture where the expected audio response from the smart device is either missing or incorrect due to server-side misinterpretation of the command. In a similar fashion to recording the data samples, interpreting the audio response from the smart device can also be partially automated. Since both a packet capture and audio response are recorded inside the dataset, we can safely associate each bad audio output with their corresponding packet capture based on the file number assigned by the recording script and the Unix modified timestamp (mtime). We have discovered that one straightforward way of automating the

[1]https://github.com/jock0rama/automated-voice-command-traffic-collection
[2]https://github.com/ACK-J/SHAME_Model_Fingerprinting_Smart_Assistants
[3]https://github.com/ACK-J/SHAME_Model_PoC

identification of bad captures is to utilize a speech-to-text engine, such as Mozilla's *deepspeech* [17].

4 ATTACK METHODOLOGY

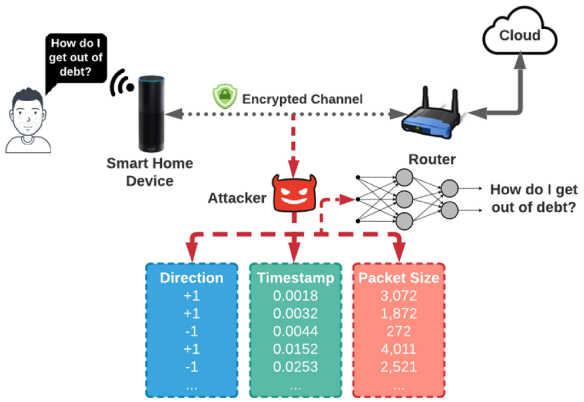

Figure 2: An attacker eavesdropping and accurately predicting the command spoken.

Figure (2) depicts how an attacker can passively sniff the traffic destined to and from the smart assistant. Even though the traffic is encrypted with TLS, side-channel information such as direction, time, and overall packet size can be extracted for offline analysis. After performing one forward pass through a trained SHAME model, the attacker will be given a prediction of what question the victim asked the smart assistant. A potential adversary could position themselves anywhere between the smart assistant and cloud server without any possibility for the victim to be alerted. Potential adversaries could include local network eavesdroppers, hostile nation-states, internet service providers, or a VPN server.

4.1 SHAME Model for VCF

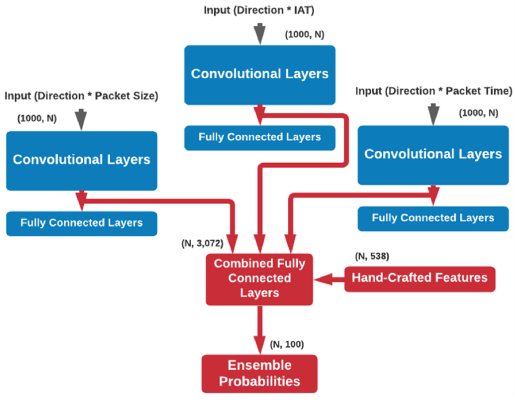

Figure 3: A deconstruction of the SHAME model.

The CNN model architecture designed by Sirinam et al. [23] was utilized due to its superior performance in the WF domain. We considered several different ways in which the voice-command traffic could be represented and fed into the network. The greatest success was found with the following three styles: *Packet Direction x Size*, *Packet Direction x Timestamp*, and *Packet Direction x IAT*. *Timestamp* refers to the relative time in milliseconds from the start of the trace, whereas *IAT* refers to the inter-packet arrival time (e.g. the time since previous packet arrival) in milliseconds.

Using only one style of representation may limit the model's depth of understanding, potentially hindering performance. For our SHAME model, we train three networks individually using the three styles of input representation. We then use the convolutional layers of each of these models as feature extractors to process samples into a new feature-vector form. These feature vectors are then used to train an MLP model to perform the final classification. To improve stability of training, we also include some handcrafted features in the feature-vectors fed to the ensembled model (the list of which is available in Appendix B). This overall scheme is described in Figure 1. In this way, the inputs to the new fully connected layers represent a large variety of features and information.

We performed hyper-parameter tuning on Sirinam et al.'s model to achieve optimal performance on our data. The most notable change we made was an increase in dropout regularization [26] within the fully connected layers of the model. Additionally, we changed the convolutional layers to use spatial dropout [27], replacing the standard dropout layers originally used (e.g. entire feature-maps are dropped, rather than individual feature outputs). For the full details of our model's architecture, consult our code [1].

5 EVALUATION

5.1 Datasets

To evaluate the SHAME model, we used the datasets collected by Wang et al. [7, 30] in addition to our smaller dataset. We captured our dynamic & static dataset [C] during March of 2021 using the processes previously described in Data Collection 3.2. Two weeks later, we were left with approximately 16,000 packet captures after filtering out erroneous captures, leaving about 800 samples for each of our twenty targeted commands. The DeepVC Fingerprinting dataset collected by Wang et al. [30] consists of 100 commands [D] with 1,500 captures each. Samples were collected separately for an Amazon Alexa and a Google Home device.

Each of these datasets represents their traffic samples as a sequence of packet metadata captured from the perspective of the routing device. The meta-data considered includes packet direction, size, and timestamp information.

5.2 Attack Performance

The accuracies obtained with each CNN model, individually, and the ensembled SHAME model accuracies are noted in Table (1). After ensembling the three models together, we observed an increase of 3.2% when trained on our Alexa dataset [C] and an increase of 2.4% on Wang et al's Alexa dataset [D]. Notably, the packet size style of representation yields the highest accuracy across all datasets. We did, however, note that using only *IAT* and *Packet Size* features was sufficient to achieve close-to maximum accuracy, indicating that much of the information contained in the *Timestamp* and handcrafted features are redundant.

Dataset	Time	IAT	Size	SHAME	DeepVC [30]
Our Alexa Dataset [C]	81.0 ± 1.2%	84.0 ± 0.7%	89.3 ± 0.8%	**92.5 ± 0.3%**	80.6 ± 8.1%
Amazon Alexa [D]	85.5 ± 3.6%	87.3 ± 0.2%	92.8 ± 0.1%	**95.2 ± 0.1%**	92.7 ± 0.4%
Google Home [D]	99.16 ± 0.02%	99.37 ± 0.05%	99.69 ± 0.02%	**99.81 ± 0.01%**	99.64 ± 0.05%

Table 1: The SHAME model performance compared to each individual model and the DeepVC [30] attack. Reported as mean accuracy with ± standard deviation over ten runs.

Table (1) shows the SHAME model's results compared to the highest scores the DeepVC Fingerprinting model was able to achieve for each dataset. When evaluating the DeepVC attack, we use the author provided code with some modifications[4]. Compared to DeepVC, our proposed SHAME model is seen to reduce the error in classifying Google Home traffic by 0.17%, achieving an almost perfect accuracy in classifying the testing samples. More significantly, the SHAME model saw a reduction in error of 2.5% when tested against Amazon Alexa traffic. Furthermore, the SHAME model has proven to greatly outperform DeepVC, reducing the error rate by 11.9%, when dynamic questions are introduced into the dataset. This could indicate that the SHAME model can generalize dynamic questions, likely thanks to our timing and directional information.

There are multiple elements of our SHAME attack that allow it to outperform the DeepVC attack. First, the architecture for our CNN models is slightly better than that of DeepVC. We compare each constituent DeepVC model with our tuned DF CNN in Appendix A and see that our model performs approximately 1% better when evaluated against Wang et al.'s Alexa dataset. Next, DeepVC ensembles multiple DL networks using a weighted sum of each model's prediction vectors. This method of joining the models limits the amount of information each model may contribute to the final classification, and it allows a poor prediction from one model to negatively affect the final output. The SHAME approach of ensembling by training a new MLP network allows for more complex classification using all the information extracted by each model. Finally, our inclusion of models trained using timing information provides a further advantage over DeepVC, as packet timing characteristics can contain critical information for correctly identifying some samples.

6 FUTURE WORK

Fingerprinting Defense. A top priority, now knowing the severity of VCF, is to develop a plausible defense that does not introduce aggressive overhead or delays. Wang et al. [30] presented a defense that implements adaptive padding [22] and differential privacy [31]. The authors achieved an impressive reduction of attack accuracy down to 28.42%. However, it seems improbable that the organizations producing these devices would willingly expend the additional bandwidth required to implement such defenses.

Real-world Evaluation. Current evaluations have considered VCF under the scope of a fairly limited number of commands. It is unclear how performance scales as the world size is increased to more closely match the real-world scope of such devices. Fortunately, the world size of VCF is relatively small when compared to other domains such as WF. Furthermore, this attack needs to be considered in the context of probable real-world attack pipelines. Sivanathan et al. [25] demonstrated that Smart Home Assistants could be identified relatively reliably by attacks outside the local network. However, the impact of errors at this stage of the real-world VCF attack pipeline has yet to be fully explored.

Data Freshness. Another question still undetermined is how voice command traffic changes over time. More specifically, the traffic fingerprint of a voice command may change due to changes in the response content (as is the case for *"dynamic"* questions) and potentially as an effect of API changes. Research in the WF domain has demonstrated that traffic fingerprints can change over relatively short periods, significantly reducing the performance of models trained on the previous *"stale"* dataset [12]. While we believe that VCF traffic is likely to remain *"fresh"* for longer periods, it is unclear to what extent this claim is true.

Uniqueness of Assistant Brands. Intriguingly, Amazon Alexa traffic seems to be consistently less finger-printable than Google devices. This may be due to slight variations in how Amazon devices communicate with the server, imperfections in the current collection of Alexa traffic, or that Amazon has implemented a subtle undisclosed defense. This oddity has proven consistent throughout the academic literature, yet it remains unclear why this is the case. Lastly, crossover effectiveness between a SHAME model trained on a specific smart assistant being reused for attacking a different branded smart assistant is unknown.

7 CONCLUSION

It is evident that VCF attacks on smart assistants such as Amazon Echo and Google Home are not only possible but can be done quickly and accurately with standard hardware by an adversary. The current assumption is that traffic patterns of commands infrequently change, leading to trivial exploitation, but these assumptions need to be validated by future works. Our research finds a significant risk to every consumer that owns a smart assistant, unknowingly disclosing private information about your personal life, device usage, and daily routine.

ACKNOWLEDGMENTS

This work was completed as a partial requirement of the Computing Security BS & MS degrees at the Rochester Institute of Technology and was funded in part by the National Science Foundation under Grants nos. 1816851 and 1433736.

[4]Modifications include changes to be compatible with Tensorflow 2.x so as to run on our system, the ability to load our dataset file formats, and an increased input data dimension size for their CNN model.

REFERENCES

[1] ACK-J. 2021. SHAME Model, Fingerprinting Smart Assistants (GitHub). https://github.com/ACK-J/SHAME_Model_Fingerprinting_Smart_Assistants

[2] Caio A.P. Burgardt Antônio J. Pinheiro, Jeandro de M. Bezerra and Divanilson R. Campelo. [n. d.]. Identifying IoT devices and events based on packet length from encrypted traffic, Computer Communications.

[3] Noah Apthorpe, Dillon Reisman, Srikanth Sundaresan, Arvind Narayanan, and Nick Feamster. 2017. Spying on the Smart Home: Privacy Attacks and Defenses on Encrypted IoT Traffic. *CoRR* abs/1708.05044 (2017). arXiv:1708.05044 http://arxiv.org/abs/1708.05044

[4] Sanjit Bhat, David Lu, Albert Kwon, and Srinivas Devadas. 2019. Var-CNN: A Data-Efficient Website Fingerprinting Attack Based on Deep Learning. *Proceedings on Privacy Enhancing Technologies* 2019, 4 (2019), 292–310.

[5] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. 2016. Hidden voice commands. In *25th USENIX Security Symposium (USENIX Security '16)*. 513–530.

[6] Batyr Charyyev and Mehmet Hadi Gunes. 2020. IoT Event Classification Based on Network Traffic. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE. https://doi.org/10.1109/infocomwkshps50562.2020.9162885

[7] Chenggang Wang et al. 2020. DeepVC Alexa and Google Home Pcap Datasets. https://drive.google.com/drive/folders/1l-fSX9VdZH5kF9z7gm82xgYX5ca0kRI0?usp=sharing

[8] festvox. 2021. Flite: A small run-time speech synthesis engine (Github). https://github.com/festvox/flite

[9] Jamie Hayes and George Danezis. 2016. k-fingerprinting: A robust scalable website fingerprinting technique. In *USENIX Security Symposium*. USENIX Association, 1–17.

[10] Jack Hyland. 2021. Exploiting Amazon Alexa Using the SHAME Model POC Video. https://drive.google.com/file/d/1nMd7PYX6JGB4ESqGlNlwv0fnka9_QMOH/view?usp=sharing

[11] Jack Hyland. 2021. SHAME Dynamic vs Static Dataset. https://drive.google.com/file/d/1K19SDZ3IdvAv_0rK6mG9d8WTpHg85gzV/view?usp=sharing

[12] Marc Juarez, Sadia Afroz, Gunes Acar, Claudia Diaz, and Rachel Greenstadt. 2014. A Critical Evaluation of Website Fingerprinting Attacks. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (Scottsdale, Arizona, USA) *(CCS '14)*. Association for Computing Machinery, New York, NY, USA, 263–274. https://doi.org/10.1145/2660267.2660368

[13] Sean Kennedy, Haipeng Li, Chenggang Wang, Hao Liu, Boyang Wang, and Wenhai Sun. 2019. I Can Hear Your Alexa: Voice Command Fingerprinting on Smart Home Speakers. In *2019 IEEE Conference on Communications and Network Security (CNS)*. IEEE. https://doi.org/10.1109/cns.2019.8802686

[14] Marc Liberatore and Brian Levine. 2006. Inferring the source of encrypted HTTP connections. 255–263. https://doi.org/10.1145/1180405.1180437

[15] Yanyan Lit, Sara Kim, and Eric Sy. 2021. A Survey on Amazon Alexa Attack Surfaces. In *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 1–7. https://doi.org/10.1109/ccnc49032.2021.9369553

[16] Jouni Malinen. 2021. hostapd: IEEE 802.11 AP, IEEE 802.1X WPA/WPA2/EAP/RADIUS Authenticator. https://w1.fi/hostapd/

[17] Mozilla. 2021. Project DeepSpeech (Github). https://github.com/mozilla/DeepSpeech

[18] Andriy Panchenko, Fabian Lanze, Andreas Zinnen, Martin Henze, Jan Pennekamp, Klaus Wehrle, and Thomas Engel. 2016. Website Fingerprinting at Internet Scale. In *Proceedings 2016 Network and Distributed System Security Symposium*. Internet Society. https://doi.org/10.14722/ndss.2016.23477

[19] Mohammad Saidur Rahman, Mohsen Imani, Nate Mathews, and Matthew Wright. 2020. Mockingbird: Defending against deep-learning-based website fingerprinting attacks with adversarial traces. *IEEE Transactions on Information Forensics and Security* 16 (2020), 1594–1609.

[20] Vera Rimmer, Davy Preuveneers, Marc Juarez, Tom Van Goethem, and Wouter Joosen. 2018. Automated Website Fingerprinting through Deep Learning. In *Network and Distributed System Security Symposium (NDSS)*. Internet Society.

[21] Nirupam Roy, Sheng Shen, Haitham Hassanieh, and Romit Roy Choudhury. 2018. Inaudible voice commands: The long-range attack and defense. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI '18)*. 547–560.

[22] Vitaly Shmatikov and Ming-Hsiu Wang. 2006. Timing Analysis in Low-Latency Mix Networks: Attacks and Defenses. In *Computer Security – ESORICS 2006*. Springer Berlin Heidelberg, 18–33. https://doi.org/10.1007/11863908_2

[23] Payap Sirinam, Mohsen Imani, Marc Juarez, and Matthew Wright. 2018. Deep Fingerprinting. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM. https://doi.org/10.1145/3243734.3243768

[24] Payap Sirinam, Nate Mathews, Mohammad Saidur Rahman, and Matthew Wright. 2019. Triplet Fingerprinting: More Practical and Portable Website Fingerprinting with N-Shot Learning. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (London, United Kingdom) *(CCS '19)*. Association for Computing Machinery, New York, NY, USA, 1131–1148. https://doi.org/10.1145/3319535.3354217

[25] Arunan Sivanathan, Hassan Habibi Gharakheili, Franco Loi, Adam Radford, Chamith Wijenayake, Arun Vishwanath, and Vijay Sivaraman. 2019. Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics. *IEEE Transactions on Mobile Computing* 18, 8 (2019), 1745–1759. https://doi.org/10.1109/TMC.2018.2866249

[26] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* 15, 1 (Jan. 2014), 1929–1958.

[27] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. 2015. Efficient Object Localization Using Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[28] Tavish Vaidya, Yuankai Zhang, Micah Sherr, and Clay Shields. 2015. Cocaine noodles: exploiting the gap between human and machine speech recognition. In *9th USENIX Workshop on Offensive Technologies (WOOT '15)*.

[29] Lionel Sujay Vailshery. 2021. Number of digital voice assistants in use worldwide 2019-2024. https://www.statista.com/statistics/973815/worldwide-digital-voice-assistant-in-use/

[30] Chenggang Wang, Sean Kennedy, Haipeng Li, King Hudson, Gowtham Atluri, Xuetao Wei, Wenhai Sun, and Boyang Wang. 2020. Fingerprinting encrypted voice traffic on smart speakers with deep learning. In *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. ACM. https://doi.org/10.1145/3395351.3399357 arXiv:2005.09800

[31] Qiuyu Xiao, Michael K. Reiter, and Yinqian Zhang. 2015. Mitigating Storage Side Channels Using Statistical Privacy Mechanisms. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (Denver, Colorado, USA) *(CCS '15)*. ACM, New York, NY, USA, 1582–1594. https://doi.org/10.1145/2810103.2813645

[32] Xuejing Yuan, Yuxuan Chen, Yue Zhao, Yunhui Long, Xiaokang Liu, Kai Chen, Shengzhi Zhang, Heqing Huang, XiaoFeng Wang, and Carl A Gunter. 2018. Commandersong: A systematic approach for practical adversarial voice recognition. In *27th USENIX Security Symposium (USENIX Security '18)*. 49–64.

[33] Nan Zhang, Xianghang Mi, Xuan Feng, XiaoFeng Wang, Yuan Tian, and Feng Qian. 2019. 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)*. IEEE, 1381–1396.

A NON-ENSEMBLE MODEL COMPARISON

Dataset	DF CNN [23]	DeepVC CNN [30]	DeepVC LSTM [30]	DeepVC SAE [30]
Amazon Alexa [D]	**92.8 ± 0.1%**	91.9 ± 0.3%	78.5 ± 0.4%	72.8 ± 0.6%
Google Home [D]	**99.69 ± 0.02%**	99.56 ± 0.05%	99.06 ± 0.06%	87.4 ± 1.3%

Table 2: Highest non-ensemble model accuracy compared across the Amazon Alexa and Google Home datasets [D].

We further examine the performance of the attack models for DF and DeepVC when they are evaluated separately from their respective ensembled models. Table 2 presents the individual model performance using the *Packet Direction x Size* input representation. This representation is the same used by the original DeepVC model. Our results show that our tuned DF CNN model is superior to the various DeepVC models when compared like-for-like. The DF CNN achieves a small performance lead of 0.9% compared to the DeepVC CNN model against their Alexa dataset.

When comparing our tuned DF CNN model with the DeepVC CNN model, we see that our model has many more trainable parameters. The DF CNN includes two additional convolutional layers that all use smaller kernel sizes. This leads to a more gradual refinement of the features layer-by-layer and is compensated for by the additional feature extraction layers. Furthermore, the usage of spatial dropout after our convolutional layers ensures that the model does not become over-dependent upon any one feature map. These elements combined produce a slightly more effective detector model.

B HANDCRAFTED ENSEMBLE FEATURES

Packet count
Byte count
Mean & median packet size
Mean, median & sum of packet sizes for intervals of 40 packets
Mean & sum of packet inter-arrival times for intervals of 40 packets
Mean & sum of packet inter-arrival times for intervals of 40 packets
Bytes per millisecond for intervals of 40 packets
Cumulative sums of packets for intervals of 10 packets

Table 3: List of handcrafted features. Traffic is organized into incoming-only, outgoing-only, and combined streams. The features are extracted for each of the three-stream types.

C DYNAMIC VS STATIC DATASET QUESTIONS

Static Questions	How many days are in September?
	How hot is the sun?
	How deep is the indian ocean?
	How far away is the moon?
	How tall is the Empire State Building?
	What is the fastest animal in the world?
	Do dogs dream?
	What is the capital of Spain?
	Is a tomato a fruit or a vegetable?
	How many days in a year?
Dynamic Questions	Tell me a joke.
	What is the price of Bitcoin?
	What is the price of silver?
	What time is it?
	Give me a random fact.
	How is the dow jones doing?
	What is the price of gold?
	What is the price of Monero?
	What is the weather in Canberra, Australia?
	What is the stock price of Tesla?

Table 4: The 20 queries used in our dataset to determine if dynamic questions drastically alter the performance of the SHAME model.

The static and dynamic questions selected for our study are presented in Table 4. The categorization of a question depends on the variability of the Alexa smart device's responses. For example, asking *"How deep is the Indian Ocean?"* will result in the same answer every time, and so is categorized as a *static* question. A dynamic question, such as *"What is the price of gold?"* will elicit different responses depending on the current price when the question is asked. We note that some questions may also result in larger variations in the response, such as *"Tell me a joke,"* resulting in shorter or longer jokes. After recording the network traces for many questions, we analyzed and selected the most observed variation as our set of dynamic questions for the full study. This should generally make for harder queries to fingerprint.

D DEEPVC FINGERPRINTING DATASET QUESTIONS

Announce Happy Valentines Day	What is gluten?
Do dogs dream?	What is Homecoming about?
Do you like cats or dogs?	What is my sports update?
Flip a coin.	What is my traffic report?
Give me a dinosaur fact.	What is on your mind?
Give me a fun fact about sleep.	What is Roblox?
Give me a patriots burn.	What is the AFC North Standings?
Good Morning.	What is the best comedy movie?
Help.	What is the capital of Spain?
How deep is the Indian Ocean?	What is the date tomorrow?
How do you spell appreciate?	What is the fourth book in the Narnia series?
How far away is the moon?	What is the history of Labor Day?
How hot is the sun?	What is the longest word?
How many days are in September?	What is the number one song this week?
How many days in a year?	What is the price of bitcoin?
How many days until christmas?	What is the scariest movie of all time?
How many days until Thanksgiving?	What is the score of the Eagles game?
How many fantasy points does LeBron James have?	What is the score of the Red Sox game?
How many ounces in a pound?	What is the time in Singapore?
How many seconds are in a year?	What is the weather for Sunday?
How many teaspoons are in a tablespoon?	What is the weather?
How much does an elephant weigh?	What is trending?
How much is an ounce of gold?	What is your favorite flower?
How old are you?	What is your favorite game?
How old is Henry Winkler?	What is your favorite hobby?
How old is Serena Williams?	What is your favorite sport?
How tall is Steph Curry?	What is your mission?
How tall is the Empire State Building?	What is zero divided by zero?
How tall is The Rock?	What movies are playing?
Is a tomato a fruit or a vegetable?	What were yesterdays scores?
Pick a number.	When does daylight saving time end?
Surprise me.	When does Game of Thrones return?
Talk like a pirate.	When is Boxing Day?
Tell me a barbecue joke.	When is Hanukkah?
Tell me a coffee joke.	When is the NBA all star game?
Tell me a fun fact.	When is the next full moon?
Tell me a Halloween hack.	Where did Yoda live?
Tell me a joke.	Where is Mount Rushmore?
Tell me a palindrome.	Who do you love?
Tell me a Star Wars joke.	Who is in Mastodon?
Tell me some good news.	Who is nominated for best actor?
Tell me something weird.	Who is playing Monday Night Football?
Translate good morning to Spanish.	Who is second in the NBA Western Conference?
What are some flower shops nearby?	Who is winning the World Series?
What are the most popular books this week?	Who is your favorite author?
What are the standings in the English Premier League?	Who is your favorite poet?
What are you thankful for?	Who is your favorite superhero?
What can you do?	Who scored for the Golden Knights?
What happened in the midterm elections?	Why do leaves change color in the fall?
What is brief mode?	Will it rain tomorrow?

Table 5: The 100 queries used to construct the DeepVC Amazon Alexa and Google Home datasets.