Technische Universität Berlin

EXPOSÉ

model2regex: Detecting DGAs with Regular Expressions Generated by a Language Model

Eric Schneider
Matr. No. 365800

Chair of Machine Learning and Security
Prof. Dr. Konrad Rieck

supervised by
Alexander WARNECKE, Tammo KRÜGER

May 14, 2024

1 Introduction

Domain Generating Algorithms (DGAs) are increasingly used in botnets as part of command and control (C&C) communication. This form of malware is used to obfuscate the real server a botnet or other distributed attacks get their instructions and updates from. By using seeded random generation of domains, these algorithms create thousands of domains and contact only a small portion of them, which makes static detection and sinkholing [9] very difficult. This strategy gives the attacker a huge advantage, because to protect against it, means taking control of possible thousands of domains, while the attacker only needs to control a single short lived domain, during the execution of their attack. A better form of protection would be to detect, if a network request is towards a potential algorithmically generated domain and block the communication at the source. DGAs, however, generate domains pseudo-randomly and use different seeds from either specific dates, twitter trends, hashes or word lists. Therefore static blocklists may not be able to keep up with blocking the communication at a network level. Deep Learning approaches have shown great promises and are currently state of the art in detecting Algorithmically-Generated Domains (AGD). Machine learned models can be used to filter network traffic. Setting them up for a filter pipeline, however, may not be very simple and act as a black box where it is not entirely sure what they are filtering.
Using algorithms from the field of language processing this thesis will attempt to learn the structure of different DGA families, using that information to generate regular expressions (RegEx). Resulting in an ease in implementing filters in existing security architecture and showing a more human readable result to help understanding the structure of learned DGAs.

2 Methodology

The main methodology of this thesis will be applied research. Using currently established solutions, from the field of language processing, I will

train a language model that learns the structure of DGAs and classifies them into their individual DGA families and benign domain names. Once the structure has been learned the research will focus on the possibility to extract information from the language model and turn it into a RegEx. If this succeeds then the work will focus on evaluating if the generated RegEx can compete with the language model itself and solutions of current state of the art. These approaches will again be iteratively improved upon through research and testing of strategies for extracting and optimizing the resulting RegEx.

3 Approach

During the prototyping phase, I will use a dataset of the top 1 million domains [4] for benign data and generate AGDs through reverse engineered DGAs [1] as malicious data. During evaluation and after finalizing the thesis I will try to work with real data from the DGArchive [5] and real data from network traffic to test outside of a ”lab setting”.
Using these labels, I then learn a recurrent neural network, as shown in Figure 1, more specifically a gated recurrent unit (GRU), a long short-term memory network with gating mechanism. Instead on a word level, however, this network will work on a character based level to predict the next character in the sequence to generate the domain. The language model will turn the possible characters of our domains into word embeddings, then feed those into the hidden states and then apply a softmax to generate the probability distribution of the next letter in the sequence. Additionally I am using the output of the last hidden state to feed the semantic meaning into a feed forward network to classify the input, as shown in Chapter 9 of the book *Speech and Language Processing* by Jurafsky and Martin [6].
After training the language model the next step is to generate regular expressions. This will be done by using the distribution output by our model after classifying the domain input. The current idea is to use specific probability thresholds to determine what RegEx atoms will be generated for

each position in the resulting expression.

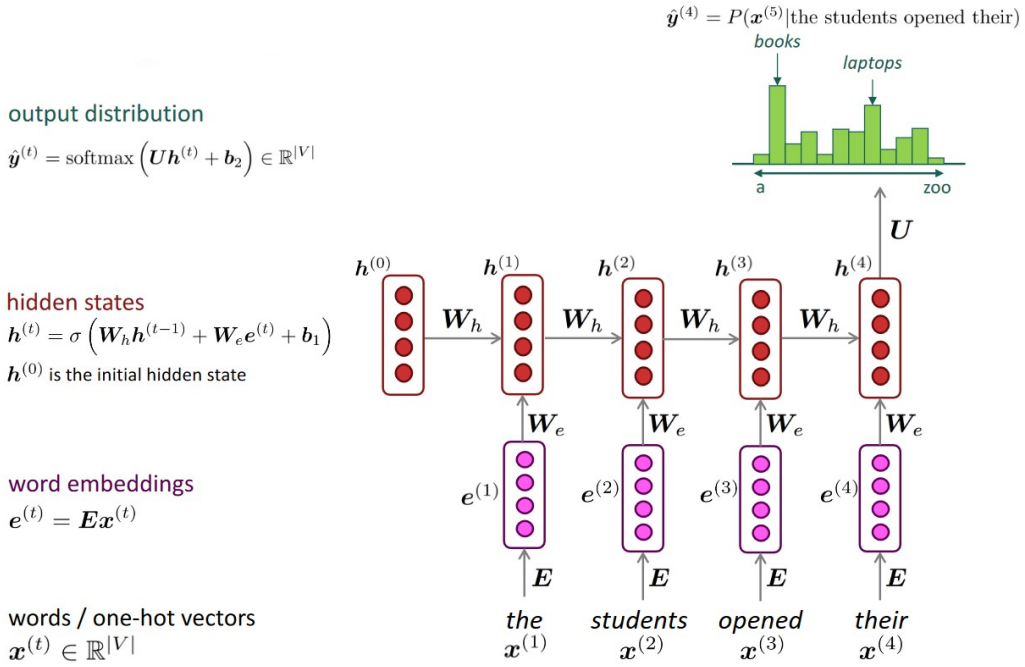

Figure 1: Example of a RNN Language Model by [7]

4 Evaluation

I will evaluate the result of this thesis by testing how well the generated regular expressions will capture the learned structures of DGAs. The experiment will be set up the following way:
The test dataset will be constructed from DGArchive[5] as the source of AGDs and real network data as the source for benign domains. Using the dataset I will compare the language model and the RegEx to the NYU model provided by Yu et al. [10], which was adapted from Zhang et al. [11]. The evaluation will compare Precision, Recall, F_1-score and area under the ROC curve (AUC), between the character based models, my own language model and the generated regular expression(s). I'll be using a single model from that paper because all compared solutions are very

close in performance therefore I will only need to test with one to be sure that the performance is comparable with the rest of the models. During evaluation it is also important that false positive rate is kept low to avoid benign domains (real domains that are not algorithmically generated) getting blocked.

5 Scope

The main scope of this work is determining the possibility of using the generated regular expressions for filtering and how well it performs compared to my language model and the state of the art. Part of the necessary work is training a multi-task criterion for the language model and the classification of DGA and benign domains. Once the classification works well the resulting RegEx should detect the learned DGA-families correctly. Generating easily readable or efficient RegEx, so using specific counts and explicit character tokens like [abc]$\{1,4\}$ instead of .*, would be a good quality to have but is not the main focus of this thesis. The resulting RegEx also does not need to be better than the current state of the art in detecting DGAs just have a close enough performance since feasibility of the approach is the main focus of this thesis.

6 Related Work

The current state of the art in the field are Convolutional Neural Networks and Recurrent Neural Networks, which are also commonly used in Natural Language Processing. Yu et al. [10] showed that all these approaches achieved similar accuracy and performance in detecting DGAs. The paper used the Bambenek dataset [2] for AGDs and the Alexa[1] Dataset for benign domains. The models were two pure RNN based Architectures (Endgame, CMU), two CNN based architectures (NYU, Invincea) and one hybrid CNN/RNN based architecture (MIT). Using one of these models,

[1] discontinued since May 2022

specifically NYU, will be baseline to compare my solution to. Regular expression are part of the field of finite state machines therefore for our model to be able to generate them it may be helpful, to treat the next token probabilities of my language model like a deterministic state machine or as a markov chain. Therefore using solutions provided by [8] and [3] are possibly good base implementations to start with.

References

[1] Johannes Bader. *Domain Generation Algorithms*. https://github.com/baderj/domain_generation_algorithms. Mar. 2024. (Visited on 03/28/2024).

[2] *Bambenek Feed*. https://osint.bambenekconsulting.com/feeds/. (Visited on 05/11/2024).

[3] Tobias Beeh. "Transformations between Markov Chains and Stochastic Regular Expressions". Bachelor Thesis. University of Stuttgart, Feb. 2017. (Visited on 05/14/2024).

[4] *Cisco Popularity List*. https://s3-us-west-1.amazonaws.com/umbrella-static/index.html. (Visited on 04/01/2024).

[5] *DGArchive - Fraunhofer FKIE*. https://dgarchive.caad.fkie.fraunhofer.de/. (Visited on 05/11/2024).

[6] Dan Jurafsky and James H. Martin. "9.3.2. RNNs for Sequence Classification". In: *Speech and Language Processing*. 3rd ed. draft. Feb. 2024. (Visited on 04/24/2024).

[7] Christopher Manning. "Natural Language Processing with Deep Learning CS224N/Ling284". Stanford, 2024. (Visited on 04/25/2024).

[8] Christoph Neumann. "CISC 3160 Programming Languages: Converting Deterministic Finite Automata to Regular Expressions". Lecture. Brooklyn College.

[9] Daniel Plohmann et al. "A Comprehensive Measurement Study of Domain Generating Malware". In: *25th USENIX Security Symposium (USENIX Security 16)*. 2016, pp. 263–278. ISBN: 978-1-931971-32-4. (Visited on 05/11/2024).

[10] Bin Yu et al. "Character Level Based Detection of DGA Domain Names". In: *2018 International Joint Conference on Neural Networks (IJCNN)*. July 2018, pp. 1–8. DOI: 10.1109/IJCNN.2018.8489147. (Visited on 03/30/2024).

[11] Xiang Zhang, Junbo Zhao, and Yann LeCun. “Character-Level Convolutional Networks for Text Classification”. In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes et al. Vol. 28. Curran Associates, Inc., 2015.