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

April 26, 2024

1 Introduction

Domain Generating Algorithms (DGAs) are increasingly used in botnets as part of command and control (C&C) communication. Malware creators use these algorithms to generate multiple possible domains each day and then have their malware contact a small portion of them to obfuscate the real server they are getting their instructions and updates from. This tactic 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 short lived domain executing their attack. Better protection may come from blocking botnets at the source by recognizing communication with specific domains as fraudulent or rather generated by a specific DGA family. DGAs however are generated 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 may not be very simple and also can be a black box on determining what exactly the model is filtering. Using algorithms from the field of language processing this thesis will attempt to learn the structure of different DGA families trying to learn their structure and 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 DGA families or into benign domain names. Once the structure has been learned the research will focus on if it is possible to extract information from the language model and turn it into a RegEx. If this is a success then the work will focus on evaluating if the generated RegEx can compete with the language model itself and solutions of the current state of the art. Again in an iterative approach of researching and testing strategies for extracting and optimizing the resulting RegEx.

3 Approach

The general architecture of the model is depicted in Figure 1.

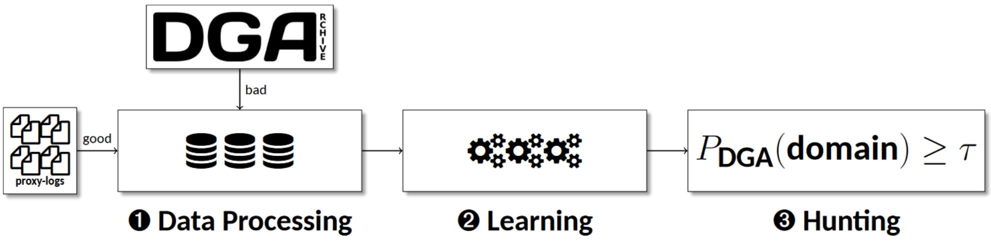
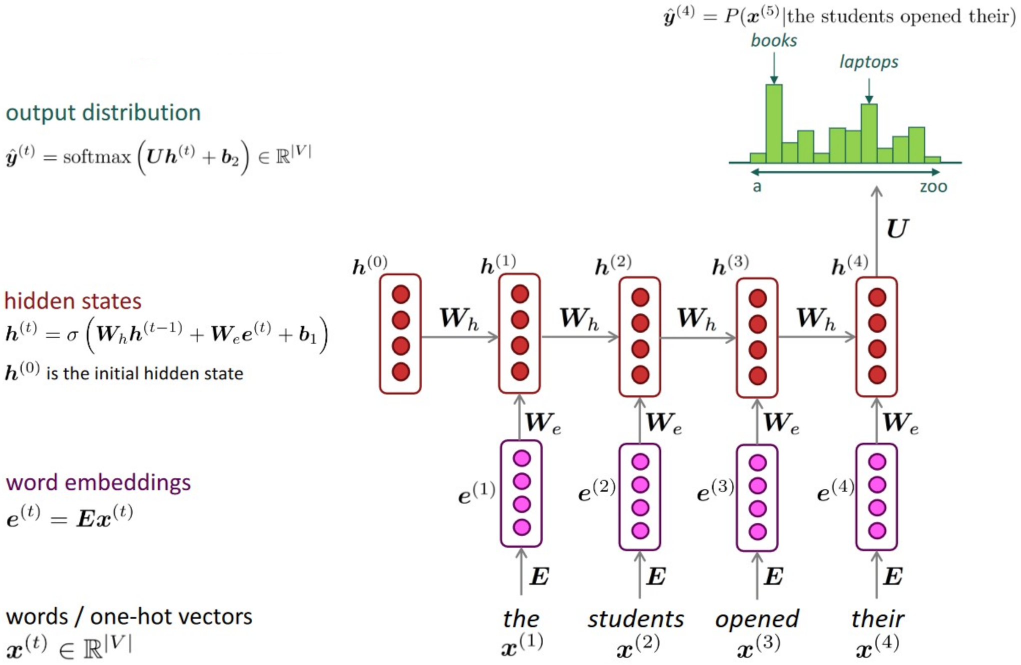


Figure 1: Graphic provided by Tammo Krüger

During the prototyping phase, I will use a dataset of the top 1 million domains [2] for benign data and generate AGDs through reverse engineered DGAs [1] as malicious data, this will be the dataset during the prototyping phase, during evaluation and after finalizing the thesis I will try to work with real data from the DGArchive[3] and real data from network traffic to test outside of a ”lab setting”. (1)
Using these labels we then learn a recurrent neural network, as shown in Figure 2, more specifically a gated recurrent unit, a long short-term memory network with gating mechanism. Instead on a word level however this network will work on character basis to predict the next character in the sequence. Additionally we are use the output of the hidden layers 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 [4]. (2)

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. (3)



$\hat{\boldsymbol{y}}^{(4)} = P(\boldsymbol{x}^{(5)}|\text{the students opened their})$
books
laptops
output distribution
$\hat{\boldsymbol{y}}^{(t)} = \text{softmax}\left(\boldsymbol{U}\boldsymbol{h}^{(t)} + \boldsymbol{b}_2\right) \in \mathbb{R}^{|V|}$
a
zoo
$\boldsymbol{U}$
$\boldsymbol{h}^{(0)}$
$\boldsymbol{h}^{(1)}$
$\boldsymbol{h}^{(2)}$
$\boldsymbol{h}^{(3)}$
$\boldsymbol{h}^{(4)}$
hidden states
$\boldsymbol{h}^{(t)} = \sigma\left(\boldsymbol{W}_h\boldsymbol{h}^{(t-1)} + \boldsymbol{W}_e\boldsymbol{e}^{(t)} + \boldsymbol{b}_1\right)$
$\boldsymbol{h}^{(0)}$ is the initial hidden state
$\boldsymbol{W}_h$
$\boldsymbol{W}_e$
word embeddings
$\boldsymbol{e}^{(t)} = \boldsymbol{E}\boldsymbol{x}^{(t)}$
$\boldsymbol{e}^{(1)}$
$\boldsymbol{e}^{(2)}$
$\boldsymbol{e}^{(3)}$
$\boldsymbol{e}^{(4)}$
$\boldsymbol{E}$
words / one-hot vectors
$\boldsymbol{x}^{(t)} \in \mathbb{R}^{|V|}$
the $\boldsymbol{x}^{(1)}$
students $\boldsymbol{x}^{(2)}$
opened $\boldsymbol{x}^{(3)}$
their $\boldsymbol{x}^{(4)}$

Figure 2: Example of a RNN Language Model by [5]

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. It is imperative that however the amount of false positives (benign domains detected as AGDs) should be very low to not block valid connections with the generated filter. The experiment will be set up the following way:
The test dataset will be constructed from DGArchive[3] as the source of AGDs and real network data will be used for the benign domains. Using

the dataset I will compare the language model and the RegEx to models provided by Yu et al. [6]. 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). During evaluation it is also important that false positive rate is kept low to avoid benign domains 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. [6] showed that all these approaches achieved similar accuracy and performance in detecting DGAs. The paper used the Bambenek dataset for AGDs and the Alexa 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). These models will also be used to compare the results of my model vs. the current state of the art.

References

[1] Johannes Bader. *baderj/domain_generation_algorithms*. original-date: 2015-08-31T09:14:32Z. Mar. 2024. URL: https://github.com/baderj/domain_generation_algorithms (visited on 03/28/2024).

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

[3] Frauenhofer FKIE. *DGArchive*. URL: https://dgarchive.caad.fkie.fraunhofer.de/ (visited on 04/26/2024).

[4] Dan Jurafsky and James H. Martin. *Speech and Language Processing*. URL: https://web.stanford.edu/%7Ejurafsky/slp3/ (visited on 04/24/2024).

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

[6] Bin Yu et al. "Character Level based Detection of DGA Domain Names". In: *2018 International Joint Conference on Neural Networks (IJCNN)*. ISSN: 2161-4407. July 2018, pp. 1–8. DOI: 10.1109/IJCNN.2018.8489147. URL: https://ieeexplore.ieee.org/abstract/document/8489147 (visited on 03/30/2024).