Application of NLP within the context of Domain Generation Algorithims and Malware

Anonymous ACL submission

1 Introduction

Command and Control (C2) centers are registered domains leveraged by threat actors such as Cybercriminals or Advance Persistent Threats(APTs), to communicate with malware-infected devices (bots). Communication between the malicious parties through Command and Control Centers allows for the exfiltration and exchange of information such as bot instructions, intelligence, telemetry, intellectual property, or other forms of data.

To create Command and Control Centers, threat actors use Domain Generation Algorithms (DGAs) to generate a large set of domains. The adversary then selects one or many domain names to register, while malware-infected devices send Domain Name System (DNS) queries to domain names within the list randomly until one of the queries resolves.

Security researchers must register the malicious domain or send takedown requests to the registrar in question to secure the threat vector, but they need to find the domain name(s). Finding the domain name(s) of command and control(s) is difficult given domain fluxing. Domain fluxing is where threat actors constantly change the command and control center domain to evade detection.

To assist in detection, various classes of Domain Generation Algorithms possess a trigger. The trigger to create a new set of domain names could be anything from time to location to even trending topics on social media. A good trigger helps the bot operator and the botnet to remain hidden, while researchers or customers can not crack the code. On top of the obscurity of the trigger, identifying a pseudo-random or word-based domain name that may or may not be lexicographically similar to other safe domains adds another layer of complexity.

2 Goal

The character-rich nature of cybersecurity has many problem areas where Natural Language Processing (NLP) can extract valuable information and be of great utility. In the context of this paper, the detection of DGA domains aid in the practical, widespread problem of malware distribution. Sub-goals for the research include:

1. training a classifier to identify known DGAs
2. generate hybrid-variants of DGAs to detect unknown DGAs

The last sub-goal is an application of natural language generation to malware, which hypothetically can increase the search coverage for new DGAs and extract code structure intent for DGAs.

Since 2015, various research groups have applied NLP to DGA detection. Highnam et al[1] created Bilbo the fibagginfi model, a hybrid neural network capable of analyzing domains by leveraging a long-term short-term memory (LTSM) and convolutional neural network for DGA Detection. The group analyzed common substrings within the domain name to ensure the robustness of the model when given a word-based domain. Word-based domain names are lexicographically similar to popular websites. Examples may include instegram.com, myface.com, or georgialech.edu. Koh et al [2] focused on context-sensitive word embeddings and leveraged ELMo to assist DGA classification. Their model architecture took less than 10 epochs to converge on consumer-grade technology, which increased my confidence in building a model to perform the same task with limited compute resources.

3 Plan

Thankfully, many researchers, such as Highnam et al.[1] and Koh et al.[2], have produced works

explaining the application of NLP within this problem area. On top of previous research performed, malware analysis is a vast, old field. A Myriad of datasets exists on Kaggle, such as:

- Microsoft Malware Classification Challenge[3] - Dataset including more than 20k examples of dissambley and bytecode
- UMUDGA - Domain Generation, https://www.kaggle.com/slashtea/domain-generation-algorithm - Dataset including 30 million manually labled DGA domains
- Domain Generation Algorithm, https://www.kaggle.com/saurabhshahane/domain-generation - Dataset including websites collected from Alexa wesbtie ranking blacklist of previous DGA domains

All of the data above will not be used but will serve as a context setter to properly invigilate the problem area at hand.

Rough Project Timeline Weeks 1-2

- Pre-proceessing and analysis of the datasets

Weeks 3-7 weeks

- Architecturing the model for classifiying common DGA Domains
- Training and Evaluating the model for classifying common DGA Domains
- Architecturing the hybrid languge generation model for two classes of DGA Domains
- Training and Evaluating the hybrid generation model

Weeks 7-8

- Analyzing and formaizling findings into a report

3.1 References

References

Kate Highnam, Domenic Puzio, Song Luo, and Nicholas R Jennings. 2021. Real-time detection of dictionary dga network traffic using deep learning. *SN Computer Science*, 2(2):1–17.