DGA Detection With Combined Machine Learning Method

Samuel Lo
glo2@illinois.edu

ABSTRACT

Domain Generation Algorithm(DGA) has been used widely by malware to establish command and control(C2) communication with infected machines. In order to prevent such communication, security organizations have to gather a list of known malicious domain to block. However, this process is tedious since DGA domains could be produced in a much higher rate. Therefore, a domain classifier is needed to solve this problem. Many of today's solutions leverage on both neural network and DNS analysis to identify indications for a DGA domains. However, neural network based design is trained based on looking at the domain name while DNS analysis is based on DNS information that could potentially be altered by attackers. Previous work exhibits poor performance only certain type of DGA domains. In this paper, we proposed a new hybrid type of classifier that combines both neural-network and DNS analysis to complement the weakness of each approach. Results are significantly better than current design of domain classifier in terms of classifying dictionary-based DGA domains and general DGA domains.

KEYWORDS

Domain Generated Algorithm, DNS Passive Analysis, Deep learning

ACM Reference Format:
Samuel Lo. 2019. DGA Detection With Combined Machine Learning Method. In *Proceedings of CS460*. 6 pages.

1 INTRODUCTION

A lot of malware today tries to establish command and control(C2) communication with infected machine. A naive way is to hardcode those IP address or domain names of those infected machine but that will make detection of traffic too trivial. Domain name could be simply blocked in order to prevent C2 communication. Therefore, malware uses domain generated algorithm to generate random domain names in order to make detection of malicious domain hard. In this way, it will be hard to detect a malicious domain because they could be easily generated in a batch. In order to detect these malicious domain names, a domain name classifier can be designed to achieve this goal. Current solutions provide different way to design these classifiers through both passive DNS analysis or neural network.

We proposed a novel way of designing this classification that combines methods from different researches. By using all the information from analyzing domain names, we are able to construct a much accurate learning models to perform this classification. Currently our classifier uses a neural network to classify domain names and combine the result from the neural network with other features obtained from DNS analysis to perform a final classification.

In summary, this paper makes the following contributions:

- Higher accuracy on detecting dictionary-based DGA domains.
- A proof-of-concept demonstrating combined machine learning method can classify all kinds of domain names, including dictionary-based DGA domains

2 BACKGROUND

Many researches have been dedicated to design a domain name classifier to separate benign domains from malicious domains. Classifying domain names is inherently a machine learning problem

CS460, Spring 2019, Champaign, IL
2019.

that could be solve using different existing machine learning algorithm. There are two major approaches to extract the features of domain to use machine learning algorithm.

2.1 Manual Feature Extraction

Manually feature extraction extracts the feature through DNS analysis to obtain information including number of distinct IP addresses or time-to-live value. The classification is determined by performing a logistic regression on all of these features. This allows more indications of whether this domain is benign or malicious. However, if the attacker is aware of these features, they could choose to implement their attack by purposely avoiding those detection. Therefore, it will be essential to construct features that the attackers could not easily circumvent it and that is one of the main challenges.

2.2 Automatic Feature Extraction

On the other side of the spectrum, another approach would be automatic feature extraction using deep neural network like convolution neural network(CNN) or long-short term memory neural network(LSTM) to classify domains. CNN and LSTM have been used widely in many machine learning problems and is known for their ability to detect spatial locality of the features. By performing CNN or LSTM on the domain names, it can easily detect gibberish domain names from benign domain names. DGA tends to generate gibberish domain names to avoid conflicting domain names with existing domain names. Therefore, these neural network is well suited for these domain names. However, there are also domain names which are dictionary-based. This means the domain name is composed of several human-readable words. Neural network performs poorly on these dictionary-based domain names because they could be easily confused with benign domain names.

2.3 Related Work

Many of the researches adopted either manual feature extraction or automatic feature extraction. Bilge et al. proposed several features obtain from passive DNS that can be used an indication of malicious domain[4]. They proposed different feature sets including time-based features, DNS answer-based features, time-to-live(TTL) value-based features, and domain name-based features and incorporate them in a decision tree to perform classification. As indicated in their conclusions, if the attacker is aware of these features, then it is possible for the attacker to circumvent these detection. Another similar work is done by Bambenek where the features used are MX record, SPF framework, and other DNS features[3].

Several neural network-based classifier with both CNN and LSTM was proposed to perform domain classification[6, 7].Instead of manually extracting different features from DNS lookup information, they used a neural network to train only the domain name itself. This gives a high accuracy on classifying domain names, especially with low false negative rate. However, as previously mentioned, it does not perform well with dictionary-based DGA domains.

There has been some researches which combines both technique to get the benefit from both approaches. Curtin et al[5] proposed a design that that trains domain names with LSTM neural network and combined it with side information that can be collected from passive DNS analysis. Their design is able to outperform existing approach and classify correctly on those dictionary-based DGA domains.

3 CLASSIFIER DESIGN

The section presents our classifier design that incorporate both neural network and passive DNS analysis for classification. In this section, we describe the design choices in detail. The main component includes:

(1) In 3.1, we describe our classifier architecture in depth.

(2) In 3.2, we describe the DNS features we chose to feed into our classifier.
(3) In 3.3, we describe our choice of neural network structure and its design space.

3.1 Classifier Architecture

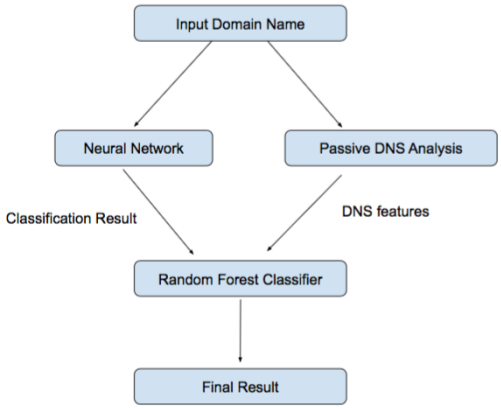

Figure 1: Classifier Architecture

Our classifier is structured as shown in Figure 1 where the input domain will be processed by two major components, neural network and passive DNS analysis. The decision output of the neural network is a boolean value indicating whether this domain is a DGA domain and the output of the passive DNS analysis is a vector of features containing DNS features. The output of the two components will be concatenated and feed into a random forest classifier to perform a final classification on whether this input domain is a DGA domain. This applies to both the training phase and validation phase. During training phase, the neural network would perform classification on the training data as part of the input for random forest classifier. Similar process happens during validation phase where neural network would perform classification on validation data set.

3.2 Passive DNS analysis

For our choice of DNS features, we select the following DNS features:

(1) Top level Domain
(2) MX record
(3) SPF record
(4) NXDomain
(5) Average TTL

These features are selected based on the result from Bambenek's research[3] and research from Bilge et al[4] since these features are proven to be good indicator of DGA domains. The additional top level domain and NXDomain to explore the possibility of additional features. For top level domain, it is one-hot encoded into numerical numbers. The other features are obtained by additional DNS queries and they are all stored in boolean value except average TTL. The boolean values essentially tells us whether this domain have this record. In the case of non-existant domain(NXDomain), all the boolean value will be false and average TTL will be 0.

3.3 Neural Network

We chose convolutional neural network and long short-term memory neural network as our neural network for automatic feature extraction. Since both neural network are being proved to yield high accuracy on classifying domains from other researches[5–7]. The input to the neural network is vectorized domain name and it is padded to 63 characters. The output will be a boolean decision of whether this domain is a benign domain. The neural network is implemented in Keras and the model is as following:

3.3.1 CNN model

```
model = Sequential(name='Seq')
model.add(Embedding(256, 128, input length=63))
model.add(Conv1D(1000, 2, padding='same',
  kernel_initializer ='glorot normal', activation ='relu'))
model.add(Dropout (0.5))
model.add(Flatten ())
model.add(Dense(100, activation ='relu',
  kernel_initializer ='glorot_normal'))
model.add(Dense(1, activation ='sigmoid',
  kernel_initializer ='glorot_normal'))
model.compile(loss='binary_crossentropy',
  optimizer='adam', metrics=['accuracy'])
```

3.3.2 LSTM model

```
model = Sequential(name='Seq')
model.add(Embedding(256, 128, input_length=63))
model.add(LSTM(units=128, unroll=True))
model.add(Dropout(0.5))
model.add(Dense(1))
model.add(Activation ('sigmoid'))
model.compile(loss='binary_crossentropy',
  optimizer='rmsprop', metrics=['accuracy'])
```

4 EVALUATION

The classifier is implemented using python3 with Keras and sklearn library for neural network and random forest classifier.

4.1 Input Set

The input datasets are 20,000 benign domains randomly chosen from Alex top 1 million[2] and 20,000 known DGA domains randomly chosen from OSINT DGA feed of Bambenek Consulting[1]. 20,000 domains from the input datasets, with 10,000 DGA domains and 10,000 benign domains, are used as training datasets. The rest are used as validation datasets, which we will denote as general validation dataset. In addition, to test whether our classifier performs well on dictionary based-DGA domains, we compiled a list of 2000 DGA domains from Suppobox, Matsnu, Pizd malware family, which is denoted as Dictionary-based validation datasets. These specific malware family are known to be dictionary-based which can create high false positive rate.

4.2 Metrics

For performance evaluation, the main metrics are classification accuracy, false positive rate, and false negative rate. The formula for each of them is shown below:

$$Accuracy = \frac{\sum TruePositive}{\sum TruePositive + \sum FalsePositive}$$

$$FPR = \frac{\sum FalsePositive}{\sum FalsePositive + \sum TrueNegative}$$

$$FNR = \frac{\sum FalseNegative}{\sum FalseNegative + \sum TruePositive}$$

Classification accuracy needs to be as high as possible for ideal classifier. Since we are interested in how our classifier will performing with dictionary-based DGA domains, false positive rate needs to be low on general validation dataset. Generally, one concern for domain classifier is the worry that benign domains will be blocked and caused disruption. Therefore, we aim to have low false negative rate in order to achieve this goal. For dictionary-based validation dataset, since it only consists of DGA domains, the only metrics that can be measured is false positive rate.

4.3 Evaluation Model

For performance comparison, we will be using classifier that consists of only one of CNN, LSTM, and RF classifier. This gives us three different base cases for comparison. For our classifier, we proposed CNN+RF classifier as well as LSTM+RF classifier.

5 RESULT

The main motivation for our classifier is to get low false positive rate on dictionary-based validation dataset and high accuracy with low false negative rate on the general validation dataset.

5.1 Baseline Classifier Performance

Figure 2 and figure 3 shows the comparison between each classifier. For accuracy on general validation dataset, there is no discernible difference between each classifier and they all achieve high accuracy. However, random forest classifier has the highest false negative rate and pure neural network-based classifier has much lower false negative rate. This is consistent with what we expected since neural network performs well on classifying DGA domains from a set of benign domains.

However, in figure 3, it is clear that both pure neural network-based classifier has significantly high false positive rate while random forest calssifier has low false positive rate. Again, this is consistent with the results concluded by other researches where neural network-based classifier are performing bad on dictionary-based DGA.

5.2 Hybrid Classifier Performance

For our hybrid classifier, it is clear that it has the benefit from both pure neural network-based classifier and random forest classifier. Not only it has much lower false positive rate and false negative rate when classifying general dataset when comparing to baseline classifier, but also it has low false positive rate when classifying dictionary-based dataset. This suggests our new hybrid classifier is able to classify dictionary-based DGA domains while at the same time maintain high performance benefit from neural network-based classifier. In other words, our new hybrid classifier can be used to classify any DGA domains, including dictionary-based DGA domains.

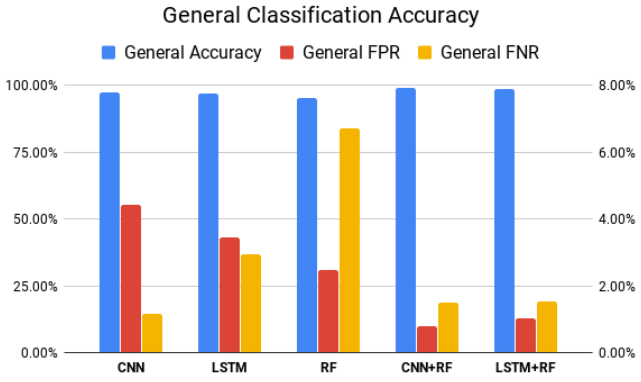

Figure 2: General Validation Dataset Accuracy

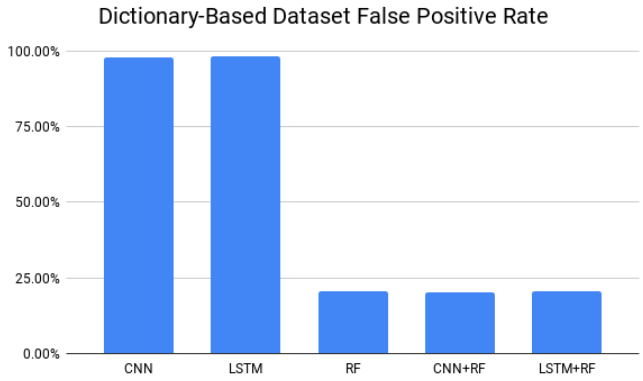

Figure 3: Dictionary-Based Validation Dataset FPR

6 CONCLUSION

In this paper we proposed a hybrid classifier with both neural-network component and passive DNS analysis component. It combines the benefit from both approaches and is able to detect all kinds of DGA domains.

One of the limitation of this paper is that the input set may not be as comprehensive as real-time network traffic. For example, the benign domains are taken from a fixed set of domains where in reality, there could be much more benign domains that are less well-known. Also, our hybrid classifier would still have the similar limitation as in the manual feature extraction approach where the attacker could attempt to circumvent the detections by modifying DNS information.

Follow-up work would include explore more different types of DNS features that could not be easily altered by attacker but at the same time being a good indicator of DGA domains. Another follow-up work would be constructing a neural network to learn about these DNS features to further improve accuracy.

ACKNOWLEDGMENTS

We would like to thank John Bambenek for his valuable advice throughout the project.

REFERENCES

[1] Bambenek consulting - master feeds. http://osint.bambenekconsulting.com/feeds/. (Bambenek consulting - master feeds). Accessed: 2019-03-14.

[2] Does Alexa have a list of its top-ranked websites?. https://support.alexa.com/hc/en-us/articles/200449834-Does-Alexa-have-a-list-of-its-top-ranked-websites. (Does Alexa have a list of its top-ranked web- sites?). Accessed: 2019-03-14.

[3] John Bambenek. 2018. Towards Robust Machine Learning on Suspicious Domain Names. (2018).

[4] Leyla Bilge, Engin Kirda, Christopher Kruegel, and Marco Balduzzi. 2011. EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis.. In *Ndss*. 1–17.

[5] Ryan R Curtin, Andrew B Gardner, Slawomir Grzonkowski, Alexey Kleymenov, and Alejandro Mosquera. 2018. Detecting DGA domains with recurrent neural networks and side information. *arXiv preprint arXiv:1810.02023* (2018).

[6] Jonathan Woodbridge, Hyrum S Anderson, Anjum Ahuja, and Daniel Grant. 2016. Predicting domain generation algorithms with long short-term memory networks. *arXiv preprint arXiv:1611.00791* (2016).

[7] Bin Yu, Daniel L Gray, Jie Pan, Martine De Cock, and Anderson CA Nascimento. 2017. Inline DGA detection with deep networks. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 683–692.