Suspicious Domain Names Detection System

Kalyan Vikkurthi (016663257), Manidedeepya Chennapragada (016045276),

Nikhil Maylarusetty 016656393

Department of Data Science, San Jose State University

MOTIVATION

Domain Generation Algorithms (DGAs) are tools used by malicious software to hide their communication. Unlike older methods that were easily detected, modern malware, like Ransomware and Advanced Persistent Threats, use DGAs to create domain names on the fly. These dynamic domain names are generated using seed values such as dates or currency exchange rates, making it harder for traditional detection methods to catch them. Our project's main goals are to identify features in both DGA-generated and legitimate domain names and use those features to build a machine learning model that can accurately identify suspicious domains.

The importance of this project lies in combating the changing strategies of malware, which now use dynamic communication methods instead of the static IP or domain methods used in the past. By studying and understanding the pseudo-random structures of DGA-generated domain names, we aim to create a strong model that can better detect and classify suspicious domains. This contributes to the ongoing efforts in cybersecurity and helps in reducing potential threats..

BACKGROUND

As technology advances, contemporary malware like Kraken has embraced Domain Generation Algorithms (DGAs) to enhance its evasion tactics. In the earlier stages, malware employed static communication methods with a hardcoded command and control server address, exposing it to the risk of IP blacklisting. The introduction of DGAs brought about a dynamic shift, where malware now generates a constantly changing list of domain names using pseudo-random number generators (PRNGs) to avoid detection and exposure of its communication channels. Despite the dynamic nature of DGA-generated domain names, they exhibit a distinctive pseudo-random structure. Our research focuses on developing a machine learning approach to identify such generated domains by extracting lexical and network-based features. We have extracted a comprehensive set of 39 features from domain names, drawing from a compiled list of clean domains from Open Page Rank and malicious domain names from Netlab360. Our experimental results highlight the efficacy of features like masked N grams, demonstrating high accuracy in detecting suspicious domain names. This research contributes to the ongoing efforts in cybersecurity, addressing the evolving tactics employed by malware for covert communication.

LITERATURE REVIEW

The literature review on the Domain Generation Algorithm (DGA) detection presents a diverse study of methods used based on advanced machine learning techniques. Zhou et al.[2] proposed a CNN-based approach demonstrating consistent performance with an AUC, score, and accuracy, achieving an impressive F1 Score of 0.9918 and Precision of 0.9961. Their model effectively identifies potential Command and Control (CC) networks overlooked by commercial tools, showing a reliable DGA detection strategy for large enterprises. Tran et al.[3] introduced an LSTM-based framework addressing multiclass imbalance in DGA botnet detection, resulting in a notable improvement of atleast 7

Highnam et al.[4] contributed to real-time

DGA detection through their hybrid model involving Artificial Neural Networks (ANN), CNN, LSTM, and the MIT (CNN-LSTM hybrid model), known as Bilbo. The comprehensive approach demonstrated high accuracy and coverage in detecting various DGAs, achieving AUC values of 0.9946 for MIT and an F1 Score of 0.9660 for CNN and Bilbo. Nonetheless, the deep learning algorithms employed in this method require longer processing times, and the URL- based comparison to a library introduces a delay in result delivery. On the other hand, Bisio et al.[5] focused on real- time behavioral DGA detection through a machine learning approach, successfully detecting all malware variants with a low false-positive rate. However, the generalization to new or evolving DGA variants, scalability to larger networks, and adaptability to emerging DGA techniques were not explicitly discussed, posing potential limitations to its broader applicability. develop a reliable plant disease classification system. To enable these models to identify and extrapolate patterns from the data, they were subjected to extensive training on our preprocessed dataset.

METHODOLOGY

Figure 1, illustrates the steps from collecting data to performing feature extraction and various processing techniques to refine and structure the data appropriately. These steps are crucial as the data preparation is essential to ensure that the data is well-suited for the modeling phase, where we will build and train our machine learning model.

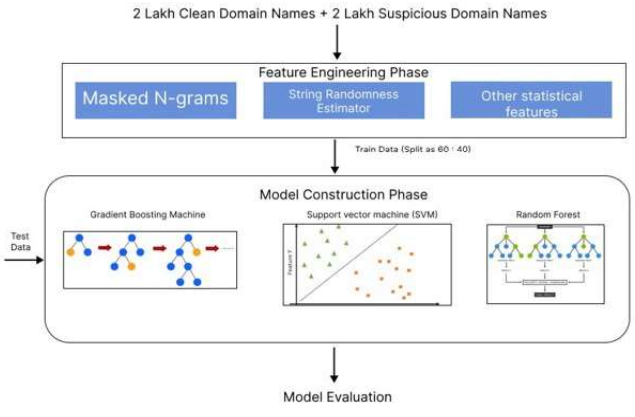

Fig. 1: Project Methodology

A. *Data Collection*

In our study, we compiled a dataset comprising 2 lakh domain names from both clean and malicious sources. This balanced dataset serves as the foundation for our analysis. The clean domain names were sourced from the Open Page Rank dataset, while the malicious domain names were obtained from Netlab360.

Rank	Domain	Open Page Rank
1	fonts.googleapis.com	10.0
2	facebook.com	10.0
3	twitter.com	10.0
4	google.com	10.0
5	youtube.com	10.0
6	s.w.org	10.0
7	instagram.com	10.0
8	googletagmanager.com	10.0
9	linkedin.com	10.0
10	ajax.googleapis.com	10.0

Fig. 2: Dataset Clean Domain

```
# DGA Domain List
# The list contains four columns:
#       DGA family, Domian, Start and end of valid time(UTC)
#
# Feed Provided By: netlab 360
# netlab@360.cn
#
# Mirai scanner daily statstics and bot IP check
#    data.netlab.360.com/mirai-scanner
# DGA domain data feed
#    data.netlab.360.com/dga
# Exploit Kit data feed
#    data.netlab.360.com/ek
# All data provided by netlab@360.cn
#    data.netlab.360.com
# About Network Security Research Lab at 360
#    netlab.360.com

nymaim  jbusvjp.org     2020-10-09 00:00:00     2020-10-09 23:59:59
nymaim  kwwouflbjf.org  2020-10-09 00:00:00     2020-10-09 23:59:59
nymaim  akjbranjwq.info 2020-10-09 00:00:00     2020-10-09 23:59:59
nymaim  xwdiklj.org     2020-10-09 00:00:00     2020-10-09 23:59:59
nymaim  siekof.biz      2020-10-09 00:00:00     2020-10-09 23:59:59
nymaim  oditgtfksm.net  2020-10-09 00:00:00     2020-10-09 23:59:59
nymaim  syhbih.com      2020-10-09 00:00:00     2020-10-09 23:59:59
nymaim  rxiucdi.com     2020-10-09 00:00:00     2020-10-09 23:59:59
nymaim  buficz.com      2020-10-09 00:00:00     2020-10-09 23:59:59
nymaim  ioryyxy.info    2020-10-09 00:00:00     2020-10-09 23:59:59
nymaim  hqratabd.com    2020-10-09 00:00:00     2020-10-09 23:59:59
nymaim  kxrgrmakda.com  2020-10-09 00:00:00     2020-10-09 23:59:59
nymaim  glszpv.biz      2020-10-09 00:00:00     2020-10-09 23:59:59
nymaim  umlddmcg.com    2020-10-09 00:00:00     2020-10-09 23:59:59
nymaim  zeditwka.org    2020-10-09 00:00:00     2020-10-09 23:59:59
nymaim  mwyvrzd.info    2020-10-09 00:00:00     2020-10-09 23:59:59
```

Fig 3: Dataset Malicious Domain

B. *Data Pre-processing*

For each domain name in our dataset, we employed a detailed feature extraction process designed to capture di- verse aspects of domain name characteristics. This involved deriving a set of features, including Masked n-gram features, String Randomness Estimator (zxcvbn score), and Other Statistical features.

i) Masked n-gram: In this step, eachdomain name was transformed into a string of symbols representing vowels, consonants, numerals, and special characters. For instance, the domain name "facebook.com" was converted to the string "cvcvcvvcscvc". Subsequently, 1-gram, 2-gram, and 3-gram features were extracted from these masked domain names. The resulting 26 masked n-gram features encompassed one-gram features ('c', 'v', 'n', 's'), two-gram features ('cc', 'cv', 'vc', 'vv', 'nc', 'cn', 'sc', 'cs', 'nv', 'vn'), and three-gram features ('ccc', 'cvc', 'vcc', 'vcv', 'ccv', 'vvv', 'cvv', 'vvc', 'ncc', 'nvc', 'csc', 'cnc'). These features aimed to capture the variations in n-gram occurrences between clean and malicious domain names, particularly crucial given that malicious domains are generated using Pseudo-Random Number Generators (PRNGs).

Features based on N-gram from the masked domain name.

Domain name	Masked domain	ccc	cvc	cc	cv	vcc	vc	v	c	vcv
facebook.com	cvcvcvvc.cvc	0	3	0	4	0	4	5	6	2
wxhyqqrbouru.pw	ccccccccvvcv.cc	6	0	8	2	0	1	3	11	1

Fig. 4: Masked N-Gram

ii) String Randomness Estimator (zxcvbn Score): Recognizing the inherently pseudo-random nature of most malware-generated domain names, we incorporated the zxcvbn score as a string randomness estimator. This score, calculated using the zxcvbn Python library, assessed the randomness of domain names. It included features such as estimated guesses needed to crack a domain name,the presence of matched English words, a numeric score (ranging from 0 to 4 indicating guessability), and a feedback warning explaining potential weaknesses. This set of four zxcvbn score-based features offered valuable insights into the predictability and structure of domain names.

Score Value	Benign eg: google.com	Suspicious eg: jbusvjp.org
Matched Word	google	NA
Score	2	4
Guesses	9470000	100000000001
Warnings	Yes	No

Fig. 5: Example of zxcvbn Score

iii) Other Statistical Features: Beyond masking domain names, we derived a set of statistical features without altering the original domain name. These included mean, variance, standard deviation for 1-gram and 2-gram values, Shannon Entropy, unique character count, and the length of the domain name. For instance, the mean for 1-gram values calculated the average occurrence of individual characters in a domain name. Unique character count and Shannon Entropy provided insights into the diversity and randomness of characters within the domain name. These statistical features contributed to a comprehensive understanding of the lexical structure of domain names.

C. *Modeling and Model Details*

For our predictive modeling, these diverse sets of features collectively formed a feature matrix for each domain name, facilitating comprehensive model training. The dataset, consisting of 39 features extracted from each domain name, was divided into training and testing sets to evaluate model performance. The models, including Random Forest, Support Vector Machine (SVM), and Gradient Boosting Mechanism, were trained to effectively classify domain names as clean or malicious based on the extracted features. The detailed feature extraction process provided the models with rich information to discern patterns and make accurate predictions, ensuring robust performance in the detection of malicious domain names.

Training Data: The dataset, consisting of 39 features extracted from each domain name, was divided into training and testing sets to evaluate model performance. The models were trained on this data, with a focus on understanding and classifying the differences between clean and malicious domain names. Training aimed to optimize the models for robust performance in detecting malicious patterns within domain names.

EXPERIMENT

Figure 1 explains briefly about the feature engineering and construction of various classification models in our work. After extracting features from the domain names (Total 4 lakh domain names: 2 lakh clean and 2 lakh

malicious), we split the data into train, validation and test splits (60-20-20). To avoid overfitting, we use a 10-fold cross validation on the training set. Below Figure 6 is the output for random forest.

```
Confusion Matrix before tuning:
 [[39074   863]
 [ 1284 38779]]

Classification Report before tuning:
              precision    recall  f1-score   support

      benign       0.97      0.98      0.97     39937
         dga       0.98      0.97      0.97     40063

    accuracy                           0.97     80000
   macro avg       0.97      0.97      0.97     80000
weighted avg       0.97      0.97      0.97     80000


Confusion Matrix after tuning:
 [[39326   738]
 [ 1346 38590]]

Classification Report after tuning:
              precision    recall  f1-score   support

      benign       0.97      0.98      0.97     40064
         dga       0.98      0.97      0.97     39936

    accuracy                           0.97     80000
   macro avg       0.97      0.97      0.97     80000
weighted avg       0.97      0.97      0.97     80000


Accuracy after tuning: 0.97395
```

Fig. 6 Hyperparameter Tuning Results

Following the work in [1]; The optimal outcomes are observed when employing the most significant features identified by the Boruta results. Specifically, we extract the top 15 features, which include metrics such as 1-gram mean, 1-gram variance, 1-gram standard deviation, 2-gram standard deviation, Unique characters count, String length, and various character combinations like 'ccc', 'cvc', 'vcc', 'vcv', 'cv', 'vc', 'cc', 'c', 'v'. Subsequently, we construct multiple classification models using these selected features, and the results are presented in Fig 9

RESULTS

Our evaluation metrics encompassed accuracy, kappa, sensitivity, and specificity, providing a nuanced understanding of the models' performance. The accuracy metric gauged the overall correctness of the model's predictions, while kappa assessed the agreement between the model's predictions and actual outcomes, accounting for chance agreement. Sensitivity and specificity offered insights into the model's ability to correctly identify positive and negative instances, respectively.

Confusion Matrix: To delve deeper into the model's performance, we utilized a confusion matrix. This matrix provided a detailed breakdown of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions. True positives represented instances correctly identified as positive, true negatives denoted instances correctly identified as negative, false positives indicated instances incorrectly identified as positive, and false negatives signified instances incorrectly identified as negative.

A. ***Results for 39 features***

The results showcased the models' proficiency in leveraging the 39-feature dataset to accurately classify domain names as clean or malicious. The inclusion of diverse features, ranging from n-gram patterns to string randomness scores and statistical characteristics, contributed to a robust representation of domain name nuances.

	Accuracy	Kappa	Sensitivity	Specificity
Random Forest	0.978150	0.95630	0.973940	0.982357
Gradient Boosting	0.971275	0.94255	0.963986	0.978559
SVM	0.974250	0.94850	0.966236	0.982257

Fig. 7: Result for 39 features

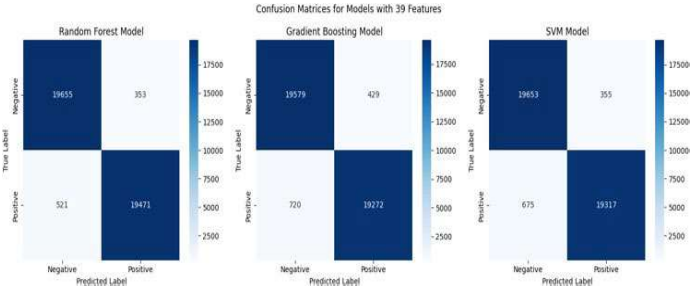

Fig. 8: Confusion matrix 39 features

B. ***Results for 15 features***

Considering the 15 features specified in the reference work by Jose[1] the construction of classifier as mentioned in the experiment section. We experimented on these features and obtained the following results where the obtained results using the 15 features showed better model performance compared to the 39 features. Thus, using these 15 features the testing domain data has been evaluated using the evaluation metrics such as accuracy, kappa, sensitivity and specificity as

shown in below figures 9 and confusion matrix is plotted as shown in figure 10.

	Accuracy	Kappa	Sensitivity	Specificit
Random Forest	0.967150	0.934302	0.961653	0.97270
Gradient Boosting	0.952025	0.904067	0.933254	0.97099
SVM	0.941625	0.883298	0.900428	0.98326

Fig. 9: Result for 15 features

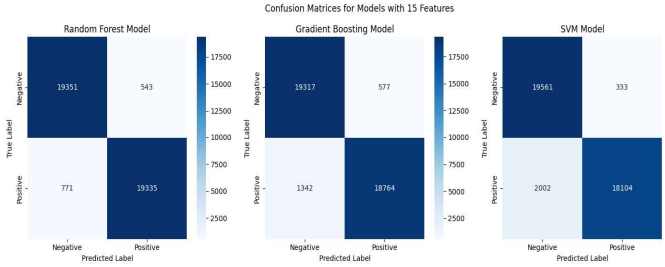

Fig. 10: Confusion matrix 15 features

FUTURE WORK

In the course of our research, we have delved into the efficacy of features such as masked n-grams, and password strength scores in constructing input vectors for our models. These features have proven highly effective in detecting variants of character-based Domain Generation Algorithms (DGAs), showcasing commendable accuracy scores. However, as we look to the future, we anticipate encountering more challenges in the realm of word-based DGAs—a relatively novel type where domain names are constructed by concatenating two or three words, departing from the pseudo-random utilization of characters typical in name generation. Word-based DGA names exhibit a closer resemblance to clean domain names, making traditional randomness-checking methods and masked n-grams less effective in distinguishing them. In response to this challenge, we recognize the imperative need to augment our feature set by incorporating more network-based features to construct a robust classifier capable of addressing the nuances presented by word-based DGAs [6][7].

Expanding on this endeavor, future work will involve a deeper exploration of network-based features, considering aspects such as traffic patterns, communication protocols, and behavioral analysis. Additionally, the integration of machine learning models with natural language processing techniques may enhance our ability to discern meaningful patterns within word-based DGAs. By broadening the feature space to include network-centric attributes, we aim to develop a more versatile and adaptive classifier that can effectively distinguish between benign and malicious domain names across a spectrum of generation methodologies. This approach aligns with the dynamic landscape of cyber threats, ensuring our models stay robust and relevant in the face of evolving techniques employed by malicious actors.

REFERENCES

[1] Selvi, Jose, Ricardo J. Rodr guez, and Emilio Soria-Olivas. "Detection of algorithmically generated malicious domain names using masked N-grams." Expert Systems with Applications 124 (2019): 156-163.

[2]“Real-time behavioral DGA detection through machine learning | IEEE Conference Publication | IEEE Xplore,” ieeexplore.ieee.org. doi: 10.1109/CCST.2017.8167790.

[3] D. Tran, H. Mac, V. Tong, H. A. Tran, and L. G. Nguyen, “A LSTM based framework for handling multiclass imbalance in DGA botnet detection,” *Neurocomputing*, vol. 275, pp. 2401–2413, Jan. 2018, doi: https://doi.org/10.1016/j.neucom.2017.11.018.

[4] “CNN-based DGA Detection with High Coverage | IEEE Conference Publication | IEEE Xplore,” *ieeexplore.ieee.org*. doi: 10.1109/ISI.2019.8823200.

[5] K. Highnam, D. Puzio, S. Luo, and N. R. Jennings, “Real-Time Detection of Dictionary DGA Network Traffic Using Deep Learning,” *SN Computer Science*, vol. 2, no. 2, Feb. 2021, doi: https://doi.org/10.1007/s42979-021-00507-w.

[6] da Luz, Pedro Marques \Botnet detection using passive DNS" Radboud University: Nijmegen, The Netherlands(2014)

[7] Bilge, Leyla, et al.\EXPOSURE: Finding Malicious Domains Using Pas-
sive DNS Analysis" Ndss 2011