BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2024
http://excel.fit.vutbr.cz

Malware Domain Detection Using Machine Learning

Tomáš Ebert

Abstract
This work aims to create a model with the best result for malware detection of domains. This is achieved by collecting as many quality malware domains as possible, and getting as much information about them as possible (lexical, DNS, RDAP, TLS), and then training and improving the classifiers to create the best model. The result of this work is a model with an F1 score of 98.05 %, which is still improving by collecting more data and tuning hyperparameters.

*xebert00@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

Today, cyber threats are becoming increasingly sophisticated and refined, posing a significant threat to the integrity, availability and confidentiality of information. One of the key threats is malware, which uses various strategies to spread and hide from security measures. Its presence can be hidden behind an innocuous-looking domain that the naked eye or "common sense" cannot detect. One great example is the domain yyoutube.com, which can be easily mistaken for the popular platform youtube.com.

The motivation for creating this work was to make the Internet safer, especially in the face of the growing threat of various cyber attacks, where, for example, older generations no longer have the ability to differentiate a bad site from a good one. However, even an experienced user can easily be fooled.

2. Related Work

Domain detection can be done in a variety of ways, one of the most well-known being the blacklisting method [1], which is known even to people who do not work in IT or cybersecurity. It works on a list of known bad domains, which were detected before.

Some researchers only focus on lexical detection. It relies on breaking down a name into subparts and finding out as much information as possible from its lexical part [2, 3]. However, this method is more suitable for URLs where there are multiple subpages, more words, etc.

Others combine lexical detection with DNS-derived information, which is much more convenient for detecting the domains themselves [4, 5].

3. Building model

This section discusses how to create a good model from data collection to training. To make it functional and compare it with the phishing model created by Adam Horák in his work, programs for the collecting and feature engineering he developed were utilized, corresponding to subsections 3.1 and 3.2.

3.1 Get External Resources

The key ingredient to getting the greatest results when training classifiers is the data. Without more data, especially good quality data, it is impossible to train a model that can stand up to a great results.

That's why the Cisco Umbrella top domains list was chosen as the benign dataset, containing the most popular and reputable internet domains.
For the malware dataset, the collection was much more challenging for many reasons. Quality blacklists are usually protected to give the company an advantage and to force people to use their antivirus. Another problem is outdated lists, where up to half of the domains can be inactive. Fortunately, there are sites like MISP, ThreatFox, URLHaus and Rescure that are updated with a new list of domains every day and it is possible to get up to 500 domains a day from all of them together. The remaining domains were collected from the public Github blacklists, tested for

their viability, tested with VirusTotal [1] to determine if it is indeed malware and then added to the collection.

3.2 Feature Engineering

To use the data for classifier training, numeric values must select from this data and encode other information using feature engineering techniques to create informative features. This is done by pipeline which converts raw data from the database to features with numerical values.

There are currently 153 features used for this model.

3.3 Model Training

The main part of the model build is the training. Lots of classifiers were used, but XGBoost was one of the best, which is why it became the main classifier for training. The first step is creating the feature vector which is used for decision-making – deciding if the domain is bad or good. The second step is finding the best hyperparameters which can help achieve better results.

K-fold cross-validation is used for evaluating the classifier's performance, providing the model's results. Because the sets are unbalanced, there are more benign than malware, the focus is on F1 (The F1 score is calculated as the harmonic mean of precision and recall, and it's particularly useful when the classes are imbalanced.).

4. Experimental results

Several models of different classifiers have been developed, but XGBoost had the best performance and is discussed in more detail. The model's F1 score came out **98.05 %** from the experimental results and other metrics like the False positive rate (false positive predictions) was 0.00385. The confusion matrix in Table 1 shows predictions on a test sample of 178,936 domains.

SHAP feature importance is shown in Figure 1. The most useful features can be considered lexical and IP features and also the RDAP. This confirms that it is a good idea to check against multiple sources and not just rely on only one type of feature like the lexical side of the domain.

		Predicted	
		Benign	*Malware*
Actual	*Benign*	138 128	530
	Malware	975	38303

Table 1. Confusion Matrix

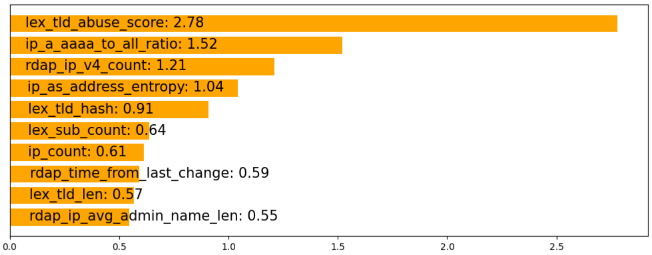

Figure 1. Features importance – SHAP

5. Conclusions

The goal of that thesis was to collect malware domains, train the classifier and get the experimental results. After the collection was done, the classifier trained, the experimental results return respectable values that can be hopefully improved even more with additional data or by tuning the hyperparameters.

In future work, the focus is on getting even better results, but also more data. An interesting area of focus would be whole URL detection, as great advantages are seen in pursuing it.

Acknowledgements

I would like to thank my supervisor Ing. Radek Hranický Ph.D. for his time, great advice and guidance throughout my work. My gratitude goes also to Ing. Adam Horák for his very well-prepared diploma thesis, which serves as a solid foundation for me to build on and extend with other types of malicious domains.

References

[1] Špaček, Martin Laštovička, Martin Horák, and Tomáš Plesník. Current issues of malicious domains blocking. *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 551–556, 2019.

[2] Wei Wang and Kenneth E. Shirley. Breaking bad: Detecting malicious domains using word segmentation. *CoRR*, abs/1506.04111, 2015.

[3] Egon Kidmose, Matija Stevanovic, and Jens Myrup Pedersen. Detection of malicious domains through lexical analysis. In *2018 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, pages 1–5, 2018.

[4] Anusha Damodaran, Fabio Di Troia, Corrado Aaron Visaggio, Thomas H Austin, and Mark Stamp. A comparison of static, dynamic, and hybrid analysis for malware detection. *Journal of Computer Virology and Hacking Techniques*, 13:1–12, 2017.

[1] VirusTotal – https://www.virustotal.com/gui/home/upload

[5] Andrej Fedák and Jozef Štulrajter. Fundamentals of static malware analysis: principles methods and tools. *Science & Military Journal*, 15(1):45–53, 2020.