Université de Namur
Faculté des Sciences
Département d'informatique

Report : Flagging suspicious hosts from DNS traces

Luyckx Marco 496283
Bouhnine Ayoub 500048

Professors: ROCHET Florentin

Course: Data analysis for cybersecurity ICYBM201

Delivered in November 2023

Academic year 2023-2024

Contents

1 Introduction

This project aims to develop a classifier that is capable of distinguishing between three different kinds of traffic : human, bot and a combination of human and bot. Based on labeled data sets, we aim to train a model to classify each host as either human or bot. This report will outline the methodologies employed, the challenges faced and the results obtained. Additionally, a critical discussion of the result will be given regarding the common pitfalls for learning-based IDS.

2 Background knowledge

Prior to the project, we had a whole lesson concerning IDS and about 2 critical aspects of IDS :

- **Effectiveness** : This dimension measures the accuracy with which an IDS **correctly** identifies intrusions within network traffic.
- **Security** : We discussed around the notion that high accuracy alone does not guarantee security. In particular, we emphasized the significance of **bayesian detection rates** and underscored the counter-intuitive nature of **the base-rate fallacy**. This fallacy suggests that even with a high overall accuracy, a system can still yield a considerable number of false positive alarms. Such a scenario is undesirable, and we must be mindful of it.

Summing up the course insights, we arrived at a fundamental understanding that, to be effective, an IDS must achieve a high level of accuracy, but to be truly secure, it must also attain a high Bayesian Detection Rate.

In addition to the course materials, we had to read further research by examining 2 articles closely linked to the project :

1. **The base-rate fallacy and the difficulty of intrusion detection**[2] : TODO demander à gépéto de faire un résumé de précis de l'article + dire que le base-rate fallacy se focus sur la partie effectiveness des 6 problèmes qu'un IDS peut avoir qui sont :
   - Effectiveness
   - Efficiency
   - Ease of use
   - Security
   - Inter-operability
   - Transparency
2. **Dos and Don'ts of machine learning in computer security**[1] : TODO demander à gépéto de faire un résumé de précis de l'article (pas oublier de mettre l'accent à fond la caisse sur les pitfalls vu que c'est ce qui nous intéresse le plus)

2.1 Pitfalls

From the second article that we read, we learn a lot about

Explain what are pitfalls in machine learning and why it is important to understand them for the project (to not fall into them, not make assumptions that are wrong, etc)

for US, from the article, the most common is : P1 : 90% P3 : 73% P10: more than 50% P9 : more than 50% P7 : more than 50%

donc c'est surement les plus probable qu'on ait dans notre projet à nous aussi, donc faut faire gaffe

2.2 Data collection and labelling

The first step for the design and development of a machine learning IDS is to obtain a comprehensive and representative dataset. From this stage, we need to consider the two following pitfalls: **Sampling bias** and **Label inaccuracy**.

2.2.1 Sampling bias

This occurs when the training data is not representative of the population it's meant to represent. In our case, we've used datasets provided by the professor (i.e., bots_tcpdump.txt and webclients_tcpdump.txt). The question that we should ask ourselves is "Est-ce que le traffic fournit par les datasets est représentatif de la pratique (qui dans notre cas sont les evaluations datasets) ? If there are not representative of the real-world distribution of human and bot DNS traffic, then our model might not perform well in practice. Since the datasets where given by our teachers, we've assumed the datasets are representative without extensive verification and that the evaluation datasets (we suppose that the professor has other datasets than from the ones that he gave us) that are going to be tried on our model are similar.

Ayoub : Est-ce que tout les hosts... The limitation of the dataset provided lies in the fact that it has a limited number of DNS traces which do not really represent the true data distribution. Indeed, these traces are restricted to a specific range of hours (within a single day) and, as a result, they do not adequately represent the whole network traffic. While efforts can be made to mitigate this bias, it is practically impossible to entirely represent the whole network behavior across all possible scenarios. Since this was given by our teachers, we cannot reduce this issue.

Additionally, the bots contained in the DNS trace are performing DNS requests from a particular botnet. Indeed, if there is another kind of botnet that reaches the network, the IDS could fail in identifying the bots since it will perform different requests. Therefore, the data will be different (It is only applicable for the case where we chose the domain as a feature !!) Therefore, they might not represent the broad spectrum of bot behaviors.

2.2.2 Label inaccuracy

Label inaccuracy refers to the incorrectness of the labels assigned to data points in a dataset. Accurate labels are crucial for classification machine learning algorithms since they rely on these labels to learn the relationship between the input data and the desired output. If it fails to do so, the overall performance of the model may be affected.

TODO je suis pas sur si ce pitfall intervient dans notre project ou pas. Vu que c'est le prof qui nous a donné les datasets, si le ground-truth des datasets est erroné, ca devient maxi problématique parce que personne ne va s'amuser à relabelisser les data (+ c'est meme pas possible imo avec tant de données)

Mais le prof s'attend peut etre a ce que l'on soit critique de ces données en disant qu'on fait l'assumption que ces données sont labelisées correctement =¿ meilleur solution selon moi

Une potentielle critique: If the labeled datasets were classified based on certain heuristics or assumptions, there might be mislabeling. For example, a host with unusual web requests that could be performed by a human might be labeled as a bot.

Additionally, if the bots change their behaviour after deploying the IDS. This could introduce a bias known as label shift (ref article Dos and Don'ts ?? pour montrer au prof qu'on a bien lu). The IDS will probably not adapt to these changes which will experience performance decay (when using in practice).

2.3 System design and learning

Now that we have the data, we need to process and use the meaningful features that are interesting for our project.

2.3.1 Data snooping

TODO, explain data snooping

checker parmi les 3 types de data snooping, lequel s'applique à notre projet et puis checkez le type voir table 8

IMO :

test snooping : - preparatory work : / (on utilise bien les données de test UNIQUEMENT pendant ls tests) - k fold cross valiation : pas sur d'avoir compris pourquoi c'est dérangeant - normalization : / (on utilise jms de fonction normalization dans le code) - embeddings : / (on utilise pas encore de neural networks donc la réponse pourrait changer) **temporal snooping** : - time dependency : oui imo, vu qu'on pense se séparer des timestamps, la gueule du traffic pourrait changer à l'avenir et notre modele devient obsolète =¿ on est dans un environnment controlé - aging datasets : / **selective snoopin** : - cherry-picking : / (on clean les datasets uniquement sur base des datasets de practices, pas de tests) - survivorship bias : /

2.3.2 Spurious correlations

Relying on relationships in the data that happen by chance. We've used domain and query type as features (and obviously others). If there are spurious correlations in our training data between these features and the labels, our model might overfit. -¿ since the majority of models are blackboxes we need to be sure that the features are well used inside the model (this can be check by reducing "unnecessary" data from the original dataset)

2.3.3 Biased parameters selection

NO, ce pitfall est non imo

We do not have this pitfall in our code since we use a separate validation/evaluation set for model selection and parameter tuning.

2.4 Performance evaluation

The way that we evaluate our model can greatly influence the outcome and lead to misleading and biased results.

2.4.1 Inappropriate baseline

To show to what extent a novel method improves the state of the art, it is vital to compare it with previously proposed methods.

2.4.2 Inappropriate performance measures

Using the wrong metric for evaluation. -¿ We've used accuracy, which might not be the best measure if our classes are imbalanced. -¿ use other metrics like precision, recall, or F1-score since there are more appropriate.

2.4.3 Base-rate fallacy

Cette section devra etre TRÈS grosse car on devra reprendre des idées du base-rate fallacy article

base-rate fallacy affects the required performance of the IDS with regard to false alarm rejection

In the training dataset, there are **343835** entries for `webclients_tcpdump.txt` whereas for for `bots_tcpdump.txt` there are only **5474**. This clearly shows that a high accuracy might be misleading.... à developper

il faut parler du false alarm rate

parler des humans aussi qui si trop de choses sont labelissés error, au fil du temps, on perd la trust du system : as the article said, "Trust once betrayed is hard to recover"

on pourrait maybe faire un Venn dessin comme pour la médical p55 de base-rate fallacy

is about the misleading interpretation of results

This pitfall has a big influence on the effectiveness parameter.

2.5 Deployment and operation

2.5.1 Lab-only evaluation

Only testing in controlled environments. -¿ Our evaluation is on the provided datasets. Real-world performance might differ.

timestamps feature is during one hour only (maybe if the bots actively perform more request during a certain period of time - outside the time that we have in our dataset - the IDS could not be able to classify it well)

If we take the hostname feature in account during the training and that during the evaluation phase, we encounter a new host that was not previously in the dataset, the model would not know how to behave.

2.5.2 Inappropriate threat model

We need to make assumptions on how we differentiate a human and a bot what are the potential data poisoning on the datasets that where given by the professor ?

pas grand chose d'autres ici je crois

3 Explain how we approach the problem

3.1 Format of DNS queries

We started by examining the structure of a typical DNS query from the tcpdumps that were given. Consider the following line : `13:22:44.546969 IP unamur021.55771 > one.one.one.one.domain 18712+ A? kumparan.com. (30)`

Breaking this down, we find :

1. **13:22:44.546969** : timestamp at which the DNS query was logged. It indicates the exact date the event occurred.
2. **IP** : Specifies that this log entry is concerning an IP packet.
3. **unamur021.55771** : This is the source of the DNS request. `unamur021` is the hostname of the device making the request and `55771` is the source port number from which the request is coming. Attention à savoir pour ne pas filter sur n'importe quoi : Port numbers are often chosen at random for outgoing requests, so it's not necessarily a fixed port for DNS queries from this device.
4. **¿** : Indicates the direction of the traffic.
5. **one.one.one.one.domain** : This is the destination of the DNS request.
6. **18712+** : This is the query ID for this DNS request. The '+' indicates that this is a recursive query.

7. **A?** : This indicates the type of DNS record being requested.
8. **kumparan.com.** : This is the domain name for which the A record is being requested.
9. **(30)** : This is the length of the DNS request packet in bytes.

When it comes to the response of a DNS query, the format remains mostly identical. However, there are some key differences :

- **X/Y/Z** : This is a breakdown of the answer sections in the DNS response:
  1. **X** : indicates there are X answers.
  2. **Y** : indicates there are Y authoritative nameservers.
  3. **Z** : indicates there are Z additional records.
- **A 104.18.130.231, A 104.18.129.231** : These are the answers to the A record query for the earlier example query `kumparan.com` and it means that the domain has at least two IPv4 addresses associated with it : `104.18.130.231` and `104.18.129.231`.

Understanding the format of these queries is essential for subsequent feature selection. We must identify features that are relevant and impactful in distinguishing between human and bot traffic.

3.2 Our assumptions

We need to put here ou assumptions :

- like with the response that appears without request - in the statement, we are looking only at DNS captures, so it will only work with DNS captures

3.3 Features selection

The development of an effective IDS based on ML requires careful consideration of the features that will be used for classification. Let's discuss the rationale behind our feature selection :

- **Timestamps**: / =¿ explain why
- **Protocol**: / =¿ explain why
- **Hostname + Source Port Number**: / =¿ explain why
- **Direction of Traffic**: / =¿ explain why
- **Destination Domain**: / =¿ explain why
- **Query ID**: / =¿ explain why
- **Type of DNS Record**: / =¿ explain why
- **Domain Name Requested**: / =¿ explain why
- **Length of DNS Request**: / =¿ explain why

Hence, we are left with X remaining relevant features and our next step involves exhaustively exploring every possible combination to identify the optimal one with the relevant machine learning algorithm (see next section faire lien hyperref).

faire graphiques pour les autres configs c-a-d des changements et alternances de features qu'on a selectionné pour voir les changements sur l'accuracy du modele + voir avec appropriate baseline

expliquer les choix de features et comment elles influencent l'accuracy mais pas que !

POSSIBLE DDoS : we get a response where we did not ask a question for a NXDOMAIN 14:55:35.387285 IP one.one.one.one.domain ¿ unamur138.47506: 10732 NXDomain 0/1/0 (120) pas de requête

3.4 Diagrams

"to be effective, the IDS must have high accuracy but to be secure, IDS must have high BDR"

- ROC curve =¿ relire base-rate fallacy p 47-48 absisse : False alarm rate ordonnée : Detection rate

absisse : false positive (en %) ordonnée : true positive (en %)

- Recall - Bayesian detection rate (BDR) =¿ relire base-rate fallacy p 44-46 (question qui me vient en lisant : comment on peut, avec les données du projet, calculer : 1. True positive rate donc le detection rate (TP) 2. False positive rate donc le false alarm rate (FP) 3. False negative rate 4. True negative rate

quand explique a ayoub, checkez fluo p 45) absisse : False alarm rate ordonnée : Bayesian detection rate

- Accuracy - Precision - comparaison entre les différents algos - comparaisons entre les différentes features - precision-recall (recall veut dire true positive rate) absisse : recall (en %) ordonnée : accuracy/precision (en %)

- F1-score =¿ aucune idée de ce que c'est mais ayoub a dit que c'est important

4 Algorithms and techniques used

In selecting the algorithms for this project, our initial step involved determining the most suitable intrusion detection strategies to implement. The seminal work on the base-rate fallacy by Axelsson [2] highlighted three major types of intrusion detection: **anomaly**, **signature** and **classical**.

We opted to focus on **anomaly detection** as our primary approach. Anomaly detection is a technique that identifies deviations from expected behavior within a system or dataset. In the context of our project, it is particularly valuable because it allows us to differentiate

between human-generated and bot-generated traffic without relying on predefined "signature" files. This is especially pertinent as we are not provided with such signature files in our dataset.

Anomaly detection, as a machine learning approach, is well-suited to handling cases where the behavior of bots may vary and evolve over time, making it challenging to create precise signatures for them. Instead, anomaly detection algorithms learn the baseline behavior of the network and can identify deviations from this norm. These deviations may indicate the presence of bots, which often exhibit patterns that deviate significantly from human traffic.

https://www.turing.com/kb/zone-intrusion-detection-with-opencv https://www.geeksforgeeks.org/intrusio detection-system-using-machine-learning-algorithms/

4.1 Algorithm 1 : Decision tree

Give explanations why we choose the decision tree has a first algo

4.2 Algorithm 2 : Logistic regression

4.3 Algorithm 3 : Random forest

4.4 Algorithm 4 : Neural networks

5 Comparison of the algorithms

6 Discussion and critics about our methodology

+ parler de ce qui pourrait etre improve

7 Conclusion

8 Bibliography

https://www.bibme.org/bibtex

References

[1] Daniel Arp et al. "Dos and Don'ts of Machine Learning in Computer Security". In: *CoRR* abs/2010.09470 (2020). arXiv: 2010.09470. URL: https://arxiv.org/abs/2010.09470.

[2] Stefan Axelsson. "The base-rate fallacy and the difficulty of intrusion detection". In: *ACM Transactions on Information and System Security* 3.3 (2000), pp. 186–205. DOI: 10.1145/357830.357849.

9 For us only

9.1 Slides

relecture du cours sur IDS key points : - haute accuracy veut pas dire bon IDS - il faut absolument qu'on intègre le BDR dans notre analyse - "to be effective, the IDS must have high accuracy but to be secure, IDS must have high BDR"

we are expected to provide visuals for the projects deadline : 8th of November. MAKE A requirements.txt at the end