Detecting Botnets Communicating with Command and Control Servers with DNS and NetFlow Data

Khanh Hoang Huynh
Security and Network Engineering
University of Amsterdam
Amsterdam, Netherlands
hhuynh@os3.nl

Mathijs Visser
Security and Network Engineering
University of Amsterdam
Amsterdam, Netherlands
mvisser@os3.nl

***Abstract*—The threat of botnets has been growing, as the number of C&C servers nearly doubled between 2017 and 2019. To defend against this threat, botnet detectors are necessary. In this research, we built a proof of concept botnet detector using DNS and NetFlow data. We make use of two independent classification systems to combine the two data sources in our proof of concept. One classification system, which we have designed, makes use of DNS and WHOIS data to determine the likelihood that a domain is malicious. The other classification system named Disclosure makes use of NetFlow data to determine if a flow is likely malicious. Disclosure was proposed by Bilge et al. [1]. Our proof of concept system combines the two scores given by the two classification systems to determine if the network traffic is malicious. We have evaluated the DNS and WHOIS features used by the domain classification system because the classification system was designed in this study. The evaluation was done by comparing the differences between benign and malicious domains for each feature. Moreover, we have also evaluated the prediction accuracy of the two classification systems individually. Finally, we have evaluated the accuracy of our whole proof of concept system. From our experiments, we have found that the chosen features of our designed classification system were properly considered. Moreover, we have found that the domain classification system and Disclosure had an accuracy of 97% and 77% accuracy respectively, given our evaluation set. Our proof of concept system had an accuracy score of 81%, given our evaluation set. We conclude that combining two classification systems is a method worth considering, despite its evaluation shortcomings.**

I. Introduction

In recent years, there is a growing trend of botnets on the internet [2]. The Spamhaus Project notes, the number of newly detected botnet Command and Control (C&C) servers had nearly doubled in 2019 compared with 2017. Botnets are used for various tasks such as executing Distributed Denial-of-Service attacks (DDoS), sending spam mail, generating fake user clicks on advertisements, and crypto mining. A botnet C&C server is used to control infected hosts that are part of the botnet. Furthermore, the document notes that cybercriminals make use of known ISPs (i.e., Cloudflare, Alibaba, and OVH) to host their botnet C&C servers. These cybercriminals are using these cloud services to hide themselves. However, these cloud services are not only used by malicious actors, thus rendering IP reputation lists useless as a defense mechanism.

A lot of research can be found on botnets and defending against them. In this research, we focus on detecting the botnets from within a network. In recent years, cybercriminals have shifted their target from individuals to businesses [3]. Therefore, in this research, we focus on detecting botnets in a corporate environment. It is important to detect and remove a bot from within a network before the bot can perform illegal activities and create unwanted traffic. A botnet can be attacked at three points: the bot, the C&C server, or the botmaster. However, these options are not always viable. In a large corporate network, it may be infeasible to remove all the bots before any harm is done to the business. Therefore, early detection and blocking traffic from infected hosts could prevent this. As mentioned earlier, botnet C&C servers may be hiding in the cloud. The C&C servers are often highly resilient and hard to target because they are a vital part of a botnet. A botmaster is also not an easy target since cybercriminals will cover their tracks carefully to avoid being prosecuted. Fortunately, a botnet can also be attacked by interfering with its network communication. Network communication is essential for a botnet to function.

To interfere with the network communication of the botnet, the bot needs to be detected first. In this research, we will look at a network-based botnet detector that relies on network behavior. Various research shows network-based botnet detectors that rely on network behavior, however, surprisingly enough, none of them combine detectors to create a more accurate bot detector. In this research, we focus on detecting botnets in a network by combining two detection systems.

A. Research questions

As we are interested in detecting malicious traffic to and from C&C servers, we have defined our main research question as such:

How can malicious traffic to and from transient Command and Control servers be detected using DNS and NetFlow data?

The following sub-questions will help us answer the main research question:

- What domain features can be used to detect transient Command and Control servers?

- What NetFlow features can be used to detect transient Command and Control servers?

B. Structure

The remainder of the paper is structured as follows. Section II gives an overview of the related work that has been done towards botnet detection using NetFlow and DNS. In Section III the necessary background information to understand the remainder of the paper is explained. Section IV gives an overview of our Proof of Concept (PoC) application, the individual components, and how they work together. Section V describes our approach of selecting and evaluating the features, as well as the experiments we performed to evaluate the accuracy of the PoC. In Section VI, the results of the experiments are described, in Section VII we discuss our findings, the limitations, and how our findings relate to existing literature. The conclusions we draw from the results are presented in Section VIII. Finally, in Section IX we suggest points for future work.

II. Related Work

As mentioned in Section I, our research uses DNS and NetFlow to detect botnets. We have compiled various related work on this topic with a summary down below.

The survey written by Zhauniarovich et al. [4] gives an overview of the current state of malicious domain detection methods using DNS analysis. In the introduction, they quickly address that the analysis of DNS data is promising to detect malicious domains. The survey describes three categories for malicious domain detection that needs to be taken into account: the data sources, approach, and evaluation. In the data source category, the paper further describes three components. The paper shows the various DNS data acquisition methods. These are, which additional information can be used from DNS data, and which ground truth sources are used. In the second category, the paper describes the different features, detection methods and the two types of outcome of a domain verdict. In the final category, the paper describes the state of evaluating the effectiveness of these domain detection systems. The paper also describes the challenges for all the categories and components. This paper was very beneficial for us as it gives a broad overview of the current state of domain reputation systems.

The paper by Mishsky et al. [5] proposed a new approach for computing domain reputation. Their solution is based on that malicious domains tend to be close to each other relational-wise compared to good domains, because a domain related to a malicious domain is also highly likely to be malicious. As stated in their paper, many research papers describe a domain reputation system for detecting botnets using DNS data. However, most of them use DNS traffic behavior and do not use the mapping information, unlike the solution proposed by Mishsky et al. and Antonakakis et al. [6].

Antonakakis et al. proposed a domain reputation system named Notos. Notos is a dynamic reputation system that gives a reputation score to new unknown domains. The system uses data from multiple sources to have up-to-date DNS information of good and malicious domains. The sources are, the DNS zone of which domain names belong to, the relevant IP addresses, BGP prefixes, AS information, and honeypot analysis. This information is used to give a reputation score to new unknown domain names. Their solution shows promising results, Notos has a high true positive rate (96.8%) and a low false-negative rate (0.36%). Initially, we had some interest in using this system as a domain classification system, given the results. Unfortunately, their additional sources of information about malicious domains and IP addresses made us create our own system. The system makes use of third-party services. This is not ideal for the PoC system we had in mind, as we did not want to depend on third-party services.

Francois et al. proposed a novel approach in botnet detection (mainly peer-to-peer) using NetFlow related data and PageRank named BotTrack [7]. PageRank is an algorithm used by Google to rank the web pages in their search engine results. BotTrack detects botnets by analyzing NetFlow data first and then create a dependency graph between hosts. The PageRank algorithm is used to extract strongly connected nodes, which are IP addresses, from the graph. BotTrack does this to find nodes that have similar roles. Also, BotTrack uses data from a honeypot to get more precision in detecting botnets. Although BotTrack shows some promising results, we opted for a different approach. For our PoC system, we wanted a system that is simple to use. BotTrack requires setting up a honeypot, which requires more work to be done to set up the detecting system.

III. Background

In this section, we explain the necessary background information to understand the remainder of the paper. The background information that we explain are the botnets, DNS, WHOIS, NetFlow, DGAs, Kullback-Leibler divergence, and the relevant vector machine.

A. Botnets

A botnet is a network of machines infected by malware. This type of malware allows a remote entity to control an infected machine. Many types of botnets and botnet architectures operate in different ways. Three main architectures exist to control the bots: centralized, peer-to-peer (P2P), and hybrid architectures [8]. In a centralized architecture, the bots only communicate to one or more C&C servers. This architecture is the most simple of the three architectures. Commonly, centralized bots use existing protocols such as Internet Relay Chat (IRC) or Hypertext Transfer Protocol (HTTP) [9]. However, the simplicity of the architecture does come at a cost. In a centralized architecture, the C&C server can be a single point of failure, shutting the C&C server down or preventing network traffic from reaching this server will prevent commands from reaching the bots. Thus, rendering the bots useless.

In a P2P architecture, botnets will not only communicate with the C&C server(s), but also form a full mesh of bots.

Every bot can communicate with other bots. However, this does come with drawbacks. As the size of the botnet increases, the number of connections required to achieve a full mesh increases significantly. Additionally, finding the initial peers and reliability distributing commands to every bot can be challenging. Because of the mentioned drawbacks, a hybrid architecture is used by more modern botnets [8].

Hybrid botnets are a compromise between the resiliency of peer-to-peer and the simplicity of centralized botnets. In a hybrid architecture, bots are divided into two groups: proxies, and workers. The worker bots do not directly connect to C&C servers to lower visibility. Instead, the workers connect through one or more proxy bots. Proxy bots are P2P connected and they pass down the information to the worker bots [8].

As mentioned earlier, communication to a C&C server is essential for a botnet. Therefore, the detection and prevention of the communication to the C&C server will ensure no new commands can be issued to the infected hosts. To block clients from reaching the C&C servers, IP blacklists could be used [10]. With the general availability of cloud services, however, these IP blacklists have become obsolete.

B. DNS

The Domain Name System (DNS) is a hierarchical and decentralized naming system that provides a mapping between domain names and IP-addresses. It is one of the backbone systems of the internet that we know today. The system is created by Paul Mockapetris in 1983, and the Internet Engineering Task Force (IETF) published two original DNS RFCs, RFC 882 and RFC 883 ([11], [12]). Nowadays, there are many RFCs which updates the DNS specifications, however, the current RFCs which describe the DNS system are RFC 1034 and RFC 1035 ([13], [14]).

DNS can be seen as a phonebook of the internet. Each entry of a phonebook is listed in alphabetical order of the subscriber name. For each subscriber name, the postal or street address and telephone number is listed. With the DNS system, the domain name is the subscriber name, and the IP-address is the postal or street name and telephone number. DNS is a client-server system, where a client will use the DNS-protocol to look up a domain name or IP-address which is answered by the DNS-server. Data in the DNS system are saved in a so-called "*resource record*" (RR). There are various types for a RR, and each type holds different information.

C. WHOIS

WHOIS is a system that is used for querying databases that store registered users or assignee of an internet resource. An internet resource can be domain names, IP-addresses, Autonomous Systems, and many more. The system has roots tracing back to 1982 when the IETF published the WHOIS protocol in RFC 812 [15]. The current WHOIS RFC is RFC 3912 [16]. Although RFC 954 [17], which obsoleted RFC 812, specify partly what information should be put in the database, it does not specify in what format the information should be saved and or displayed. Therefore, the WHOIS system is not uniform, and many variations of the data can be found depending on the provider. Nevertheless, the system holds information that can be useful for determining the intent of a domain that can be benign or malicious.

In this paper, we will focus on the WHOIS system used for domains. The WHOIS databases used for other internet resources are not used in this research, and therefore, they are not mentioned. The information of a domain name that we get from the WHOIS system is the creation date, last updated date, the expiration date of the domain, and the domain name.

D. NetFlow

NetFlow [18] is a widely used feature that enables network devices, such as routers and switches, to collect IP network traffic entering or exiting a network device. NetFlow represents IP traffic as flows. A flow is defined as a unidirectional sequence of packets that all share the following values:

1) Ingress interface
2) Source IP address
3) Destination IP address,
4) IP protocol,
5) Source port for UDP or TCP
6) Destination port for UDP or TCP
7) IP type of service.

E. Domain Generation Algorithm

A Domain name Generation Algorithm (DGA) is an algorithm used to generate large numbers of domain names. A DGA can also be used by bots to find C&C servers. The following four different DGA types are recognized by Plohmann et al.

1) **Arithmetic-based DGAs** are the most common type of DGA. This type of DGA calculates a sequence of values that can be used to represent a valid domain name.
2) **Hash-based DGAs** use the output of hash functions such as SHA1 or MD5 to represent domain names.
3) **Wordlist-based DGAs** are similar to arithmetic-based DGAs, however, instead of generating random ASCII values, this type of DGA will randomly select a word from a word list. These word lists are either directly embedded in the malware binary or retrieved from a publicly accessible source.
4) **Permutation-based DGAs** derives the possible domains through the permutation of the initial domain name. Some implementations use pseudo-random number generators (PRNGs) to generate domains.

These algorithms are used by malware to circumvent domain-based firewall rules and make it hard to shut down botnets, as a large number of new domain names can be attempted every day [19].

F. Relative Entropy

In 1948, Claude Shannon published the famous paper "*A Mathematical Theory of Communication*" [20]. Among other things, this paper introduced the concept of information entropy. Information entropy quantifies the amount of uncertainty

involved in the value of a random variable or the outcome of a random process. In other words, the measure of randomness or uncertainty in data. This is known as the Shannon entropy and is denoted by the following formula:

$$H(P) = -\sum_{i}^{n} p_i \log_b p_i \tag{1}$$

where b is the base of the logarithm used, p_i is the probability of character number i appearing in a stream of characters P, and n is the length of P. Shannon entropy works well with truly randomized data. In our case, domain names are not truly randomized data, and certain characters occur more often than others in the domain names. However, relative entropy, which compares two probability distributions could solve the problem.

Relative Entropy, also known as the Kullback-Leibler divergence [21], is a measure of how one probability distribution is different from a reference probability distribution. In our case, we would be able to measure the probability distribution of the characters in the domain name and compare it with the average probability distribution of characters of multiple domains. The relative entropy formula is denoted as:

$$D_{KL}(P_{||}Q) = \sum_{i}^{n} p_i \log_b(\frac{p_i}{q_i}) \tag{2}$$

where b is the base of the logarithm used, p_i is the probability of character number i appearing in a stream of characters P, q_i is the probability of character number i in the reference probability distribution Q.

IV. System Overview

The PoC that we have created, is able to detect botnets by making use of Netflow and DNS data. Our system consists of four main components as seen in Figure 1. The first component in our system will try to map DNS lookup results with the NetFlow data. Then we make use of two classification systems to detect botnets. These classification systems are the domain classification system using DNS and WHOIS data, and a NetFlow classification system named Disclosure. The use of two classification systems allows the PoC to make use of more data sources to classify network traffic. The final component of our PoC is the component that combines the score given from the two classification systems. The location of the PoC code can be found in Appendix A.

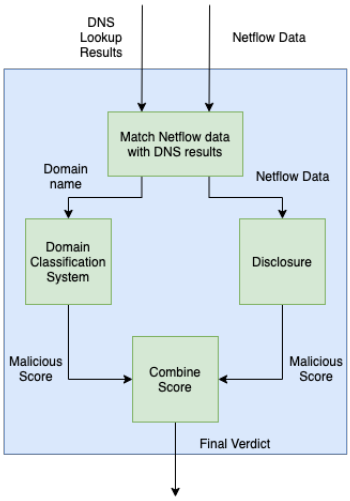

Figure 1: Proof of Concept Architecture

A. Mapping NetFlow Data with DNS

NetFlow flows have to be mapped to domain lookup results, to combine the results of NetFlow and domain classification. Since NetFlow flows do not store domain lookup results, one would have to temporarily store them. In our PoC, we created a temporary database that stores the source IP address and resulting A or AAAA DNS resource records on the DNS server. This allows us to match the source IP address with the NetFlow source IP address so that it is possible to classify both the NetFlow flows as well as the domain name on being malicious. This approach, however, does require clients to use a designated DNS server. Another approach is to capture packets on the well-known DNS port 53. This would allow clients to use other DNS servers as well. However, a prerequisite is that clients do not make use of DNS over TLS (DoT) or DNS over HTTPS (DoH). These techniques have not yet been widely implemented in operating systems. Therefore, this approach is currently still viable.

B. NetFlow Classification system: Disclosure

In 2012, Bilge et al. published a paper about Disclosure [1]. Disclosure is a wide-area botnet detection system using NetFlow data. The system uses a set of features and feeds this to a machine-learning algorithm to detect a botnet by classifying the NetFlow flows. The set of features can be split into three classes: the flow size, client access patterns, and temporal behavior. This exact system is used in our research to detect malicious NetFlow flows, and therefore detect botnets. Disclosure does this by assigning a malicious score to the flows. A flow is considered malicious if the malicious score is 50% or higher.

C. Domain Classification System

The domain classification system is the other classification system that we have used in our PoC. This classification system is designed and developed by us based on the known literature. The system takes a domain name as input and uses

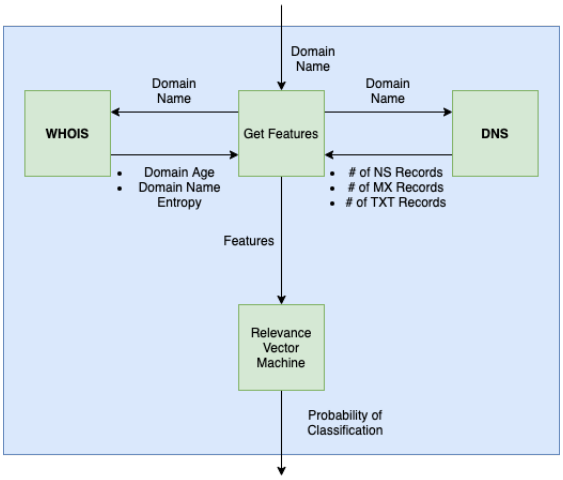

Figure 2: Proof of Concept Domain Classification System Architecture

the domain name to get its features. Once all the features are collected, the features are passed on to the Relevance Vector Machine (RVM) [22]. Consequently, the RVM will classify whether the domain is malicious or benign, given the features of the respective domain name and the trained model. This is done by assigning a probability of the classification to the domain. The probability of the classification shows the malicious score of the domain. A domain is malicious if the score exceeds 50%. An overview of the domain classification system architecture can be seen in Figure 2.

1) Features: The features of the DNS domain classification system can be split up into two categories, *DNS*, and *WHOIS*. As described earlier in Section III, DNS helps to translate domain names to IP addresses and vice versa. WHOIS is a system that shows the information on who is responsible for a domain or IP address.

The DNS is useful for classifying malicious domains [23][24]. The DNS features that were chosen are the number of NS, MX, and TXT records. These features were selected based on a paper by Kuyama et al. [25]. Additionally, we have also re-evaluated their findings. The experiments to re-evaluate the features are described in Section V, and the results of the experiment can be found in Section VI. The reason for looking only at the NS, MX, and TXT records was due to the limited amount of time. Moreover, the goal of this research was to research the approach of using two classification systems to detect botnets.

In addition to the DNS features, we also use features that use information obtained from WHOIS. The features that we use are the registration period of a domain and the entropy of the domain name. These features were inspired by the papers from Kheir et al. [26] and Plohmann et al. [19]. The registration period of a domain name is dependent on the available information of the registrars. The registration period is the expiration date of the domain minus the updated date of the domain. The registration period from malicious domains tend to have a short registration period before they are shut down compared to benign domains [27]. We have conducted an experiment that is described in Section V to verify this claim. The results of the experiment can be found in Section VI.

2) Detection Model Creation: The machine learning algorithm used for this system is the RVM. We considered a probabilistic binary classifier because it was desired to classify a domain on being benign or malicious. The probabilistic score is used to determine, how confident the classifier is, which allows us to better combine two or more scores. Moreover, the classifier should be able to handle a vast number of features and allow classification of domains in real-time. These criteria fitted well with the RVM machine learning algorithm.

A drawback of using an RVM is that one would not be able to extract the weights of the features of the trained model. It is desirable to know the weights, to understand how classification decisions are made. Not only RVM, but other machine learning algorithms also face the same problem when multiple features are involved. However, manually determining the weights of each feature is impractical if not impossible, when there is a lot of features. Thus, we still think machine learning algorithm is the better choice despite the lack of being able to extract the weights of each feature.

D. Combining the Scores

Disclosure and the domain classification system classify and give a score individually on a NetFlow flow and domain respectively. This component gives a final verdict on the network traffic by combining the score of Disclosure and the domain classification system.

V. Methodology

In this section, we describe the methodology of the performed experiments. We first give an overview of the experiments conducted. Then, we describe the experiments in more detail. The location of the files used to perform the experiments described in this section can be found in Appendix A.

As mentioned in Section IV, our PoC consists of four main components. Three components provide a score that is used to classify network traffic. The three components are the domain classification system, Disclosure, and "Combining the Scores" component. As they provide the score to classify network traffic, it is to important evaluate them.

The domain classification system is a system that we had mostly designed from scratch. The features that were used for the domain classification system were from other researches. We re-evaluated those features to see if the claims of other researchers were correct and if they were still applicable. The three features that we have re-evaluated are: domain registration period, domain resource record count, and domain entropy.

After the domain classification system its features were re-evaluated, we trained the two classifications system using a training dataset and control dataset. After the classification systems were trained, we evaluated each classification system

using an evaluation dataset. This evaluation dataset is a completely different dataset from the training and control dataset. The evaluation of each classification system was done, in order to combine the scores.

A. Domain lists

In order to perform our experiments, we had to create two lists, one list containing malicious domains and one list containing benign domains. To create the list of benign domains, we selected the Majestic top 1 million domains[1]. This is a publicly available list of domains with the largest number of referring subnets. We believe this list is unlikely to contain any malicious domains, however, this is still an assumption we had to make.

To create the list of malicious domains, we used two blacklists provided by Joe Wein. One list contains the domains that were recently used for malicious purposes, this blacklist has a retention period of two weeks[2]. The other list contains malicious domains that were outside the two weeks retention period[3]. We have created a malicious domain list containing both recent as well as older domains because the recent blacklist only contained a small number of domains.

The malicious domain lists from Joe Wein do not contain botnet C&C domains. However, due to the lack of publicly disclosed recent C&C domain lists, the list that we have chosen was the best possible alternative. The malicious domain lists from Joe Wein contain domains used for spam activities, which most likely have the same attributes as other domains used for illicit activities such as botnet C&C server. However, this is an assumption we had to make due to the lack of publicly disclosed recent C&C domain lists.

We combined 250 recent malicious domains and 250 older domains to create our malicious domain list. Consequently, we randomly shuffled this domain list to create a randomly shuffled malicious domain list of 500 domains.

The sources of the domain lists were all accessed on 1 July 2020.

B. Feature Evaluation

The selected features were evaluated to ensure that there is a correlation between the selected features and malicious or benign activities. Feature evaluation is important as the performance of the machine learning models heavily depend on distinctions between malicious and benign features. We will perform the experiments using the domain lists as described in Section V-A.

1) Registration Period: The registration period of a domain can be used to get an indication of the age of a domain. Among other features, the registration period can be used to detect whether a domain is malicious or benign. In general, benign domains are older than malicious domains [27].

The registration period is calculated by querying a WHOIS database for a domain first. The database would answer with the relevant information of the domain. There are three information fields that we are interested in, the creation date, the last updated date, and the expiration date of a domain. The creation date of a domain, show the date when the domain was created. The last updated date of a domain shows the most recent date when the WHOIS information of the domain was modified. The expiration date of a domain is the date when the domain expires from the owner unless the domain is renewed.

In general, the registration period is determined by calculating the number of days between the creation date of the domain and the expiration date of the domain. Unfortunately, the WHOIS information retrieved from the various database is not uniform, depending on the registrars. Thus, one would not be able to always find the same information fields. Thus, if it was not possible to calculate the registration period generally, we would calculate the registration period with the information that is available. For example, if the WHOIS response did not contain the expiration date and last update date, then we would calculate the registration period by calculating the number of days between the creation date and the day we are querying. Thus, for our evaluation, we calculated the registration periods of the benign and malicious domains.

We took the first 300 domains of the malicious domain list due to time constraints. We also took the first 300 domains of the benign domain list. The malicious and benign domain lists are described in Section V-A.

2) Resource Records: RRs for a domain can be directly retrieved using DNS. We only queried the root domains in both the benign and malicious domain lists. For every domain, we would query for all the RR types, as shown in Table I, at the root domain. For example, if the to be queried domain also had a subdomain, i.e., *sub.example.com*, we would only query to root domain, i.e., *example.com*. Using these RRs we can calculate which RR type has the largest difference between C&C and benign domains. These RR types could potentially be used as features to detect C&C domains.

We expect to find differences in the number of MX and TXT records, as legitimate domains will likely configure email, or verify their domain using TXT records, whereas malicious domains will be less likely to do so. To determine which resource records can be used as features, we use both the mean as well as the standard deviation. The standard deviation shows the amount of variation or dispersion of a set of values. RR types with large differences between the mean of benign and malicious domains and low standard deviations are the most suited as features.

We used the two domain lists as described in the previous Section V-B1. We queried 300 malicious and 300 benign domains for every RR type that is shown in Table I.

3) Domain Entropy: Malicious domains that have random domain names generated by DGAs, will most likely have a higher entropy score compared to benign domains. However, most domain names do have some parts in common, such as the top-level domain (TLD) and possible subdomains such as *www*. To prevent the score from being impacted by these characteristics, we will only use the actual domain name

[1]https://majestic.com/reports/majestic-million
[2]https://joewein.net/dl/bl/dom-bl.txt
[3]https://joewein.net/dl/bl/dom-bl-base.txt

excluding the subdomain(s) and TLD. Multiple algorithms can be used to calculate an entropy score for a domain name such as Shannon entropy or relative entropy, as described in Section III-F. We expect relative entropy to perform better, as certain letters are more or less likely to appear in normal domain names than in DGAs. To test this theory, we will calculate both the Shannon entropy, as well as the relative entropy of each domain to explore which type of entropy creates a more distinctive feature.

As relative entropy calculates the divergence between two probability distributions, we have to create a baseline probability distribution. The baseline probability distribution was created by using the actual domain name as described earlier with the top 1 million domains list from Majestic.

We have used the same domain lists, as described previously for the other two features. This means that we calculated the Shannon entropy and relative entropy of 300 benign and 300 malicious domains.

C. System Evaluation

We evaluate each classification system and our PoC on their effectiveness. The accuracy was measured to indicate the effectiveness of the systems. To evaluate the classification systems and our PoC, we have used one dataset, referred to as the evaluation dataset in the remainder of the paper. However, before we are able to measure the accuracy of the systems, we first had to train the classification systems. This is described in their respective subsections. We continue with describing the evaluation dataset first.

The evaluation dataset we used was retrieved from www.malware-traffic-analysis.net [28]. This website publishes unmodified packet captures of recent malicious traffic. Domain information and NetFlow data can be extracted from the published packet captures, this allows us to evaluate both systems separately and as a whole. We used all packet captures published between May 19, 2020, and June 6, 2020, as older packet captures contained C&C servers that often no longer exist. The resulting evaluation dataset had a total of 459 NetFlow flows and 359 unique domains. The difference in the number of NetFlows and domains can be explained by two possibilities. The first possibility is that multiple NetFlow flows contained the same destination domain. Another possibility is that the NetFlow flow contained an IP-addresses that had not been retrieved using DNS. Therefore, the destination domain remained unknown. We have identified the malicious flows and malicious domains using the information provided with the packet captures. Out of the 459 NetFlow flows, 72 flows were malicious. The remaining 387 flows were benign. Out of the 359 domains, 42 domains were malicious and 317 domains were benign.

The results of the evaluation of the two classification systems and the PoC will be reported in a confusion matrix and an accuracy value. In the confusion matrix of our research, a true positive (TP) shows the number of correctly classified malicious items. True negative (TN) shows the number of correctly classified benign items. False positives (FP) shows the number of incorrectly classified malicious items. False negatives (FN) shows the number of incorrectly classified benign items.

The accuracy is calculated as:

$$Acc(\%) = \frac{TP + TN}{TP + TN + FN + FP} \times 100 \qquad (3)$$

where TP is the number of true positives, TN is the number of true negatives, FN is the number of false negatives, and FP is the number of false negatives.

1) Domain Classification System: To train our domain classification system, we used the two domain lists as described in Section V-A. We randomly took 400 malicious and 400 benign domains from the benign and malicious domain lists. These were then combined and shuffled to create the training set. This training set contains a total of 800 domains. This training set is used to train our domain classification system.

The remainder of the malicious and benign list were then combined and shuffled to create our control set. This control set contains 100 malicious and 100 benign domains. The control set was used to measure the accuracy based on the training set. This accuracy only gives us insight into whether we are under- or overfitting or not.

After we have trained the domain classification system using the training set and verified the trained model with the control set, we evaluated the model using the evaluation dataset. The domain classification system only uses the 359 unique domains of this evaluation dataset. In the results, we show the confusion matrix as well as the accuracy, shown in Equation 3, of the domain classification system.

2) NetFlow Classification System: Disclosure: To train NetFlow classification system, we used a CTU-13 dataset [29]. We chose scenario number 43 of the CTU-13 dataset as it contains both raw packet captures, as well as labeled bidirectional NetFlow data of both botnet and benign traffic. The dataset has been captured in a university network in 2011. The complete dataset contains over 6 million individual NetFlow flows, of which 54433 NetFlow flows were labeled as malicious. We took 80% of the total malicious NetFlow flows, and combined it with an even number of randomly chosen benign NetFlow flows to create the training dataset for the NetFlow classification system. The training set consists of 43546 malicious and 43546 benign NetFlow flows, resulting in a total of 87092 NetFlow flows.

After we have created the training dataset, we created the control set. The remaining 10887 malicious NetFlow flows were combined with an even number of randomly chosen benign NetFlow flows to create the control set. The resulting control set for Disclosure had a total of 21774 NetFlow flows that had an even number of malicious and benign NetFlow flows. We made sure that the benign and malicious NetFlow flows in the control dataset were different from the training dataset.

Once the NetFlow classification system was trained, we evaluated the model with the evaluation dataset. This classification system only uses the NetFlow data from the evaluation

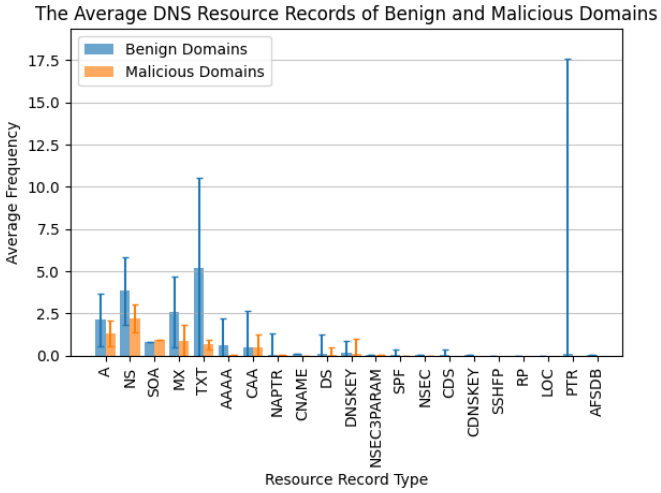

Figure 3: The Average DNS Resource Records of Malicious and Benign Domains

RR type	Mean Benign	Mean Malicious	Mean Diff.
A	2.118	1.324	0.794
NS	3.830	2.199	1.631
SOA	0.830	0.953	0.123
MX	2.569	0.848	1.721
TXT	5.209	0.657	4.554
CNAME	0.110	0.068	0.042
AAAA	0.618	0.459	0.159
PTR	0.102	0.020	0.082
SPF	0.027	0.010	0.017
DS	0.104	0.034	0.070
DNSKEY	0.181	0.095	0.086
NSEC3PARAM	0.022	0.027	0.005
NSEC	0.052	0.007	0.045

Table I: Mean Number of Resource Records Type for Benign and Malicious Domains in our Evaluation Dataset

RR type	Mal. SD	Mal. RSD	Ben. SD	Ben. RSD
A	0.781	59.0%	1.541	72.8%
NS	0.809	36.8%	1.995	52.1%
SOA	0.000	0.0%	0.000	0.0%
MX	0.989	116.6%	2.103	81.9%
TXT	0.279	42.5%	5.341	102.5%
CNAME	0.000	0.0%	0.000	0.0%
AAAA	0.788	171.7%	1.583	256.1%
PTR	0.000	0.0%	17.500	17156.9%
SPF	0.000	0.0%	0.314	1163.0%
DS	0.433	1273.5%	1.152	1107,7%
DNSKEY	0.884	930.5%	0.692	382.3%
NSEC3PARAM	0.000	0.0%	0.000	0.0%
NSEC	0.000	0.0%	0.000	0.0%

Table II: Standard Deviation and Relative Standard Deviation for Malicious and Benign Domains per Resource Record Type in our Evaluation Dataset

dataset. Thus, only 459 NetFlow flows will be classified. The results are shown in a confusion matrix. Furthermore, the accuracy is calculated as shown in Equation 3.

3) Proof of Concept: Our PoC combines the two trained classification systems as described in Section IV-D. The component that gives the final verdict of the PoC is the "Combining the Scores" component. This component takes the accuracy of the two classification systems into account.

We evaluated the PoC once all components were finalized. This was done by using the evaluation dataset. The evaluation dataset contains both 359 unique domains and 459 NetFlow flows. We show results in a confusion matrix. Additionally, the accuracy is determined as shown in Equation 3.

VI. Results

In this section, we show the results of the feature evaluations, as described in our methodology. Furthermore, this section also shows the evaluation results of the two classification systems and PoC system.

A. Feature Evaluation

1) Domain Resource Record Count: Figure 3 shows the mean amount of each RR type of malicious and benign domains. It shows that the number of NS, MX, and TXT RRs differ the most between malicious and benign domains. Additionally, it presents the standard deviation for every record type. Table II shows the measured values more precisely, as well as the mean difference between malicious and benign domains.

The average of each RR type of benign domains was an important factor in evaluating and choosing the features. RR types where its average number for benign domains was lower than 1 were disqualified to be used as a feature. The reason for that is, not all benign would have at least one RR of that RR type on average. This shows that the RR type is not a distinctive feature of benign domains. The remaining RR types that we could use as features were A, NS, MX, and TXT RRs. Additionally, as previously mentioned in Section V-B1, we also look at large differences between the mean number of RR types of benign and malicious domain. In Table I, we can see that RR types, NS, MX, and TXT had an average difference of 1.631, 1.721, 4.554 respectively. RR type A also had a relatively large mean difference. However, as its mean difference was below 1, we concluded that this was not distinctive enough. Furthermore, we also looked at the average relative standard deviation of benign and malicious domains of each RR type. These values can be found in Table II. The RR types were ignored where the standard deviation for either malicious or benign domains is 0%, because they already had a low mean difference. AAAA, DS, and DNSKEY RRs had too high of a standard deviation, making it unreliable to be used as a feature. The RR types that could be used considering the relative standard deviations are RR type A, NS, MX, and TXT.

In conclusion, considering the results in Table I and Table II, the number of RR type NS, MX, and TXT are most likely good features.

2) Domain Registration Period: As described in Section V-B1, we have calculated the registration period of 300 malicious and 300 benign domains. Figure 4 shows the frequency of the registration days of malicious and benign domains.

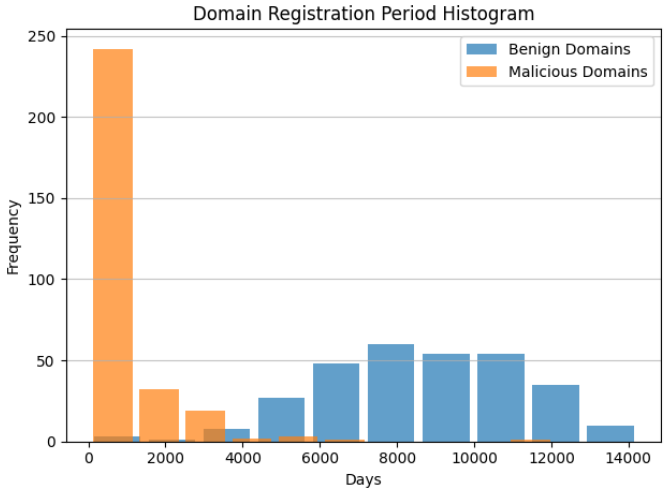

Figure 4: The Registration Periods 300 Malicious and 300 Benign Domains

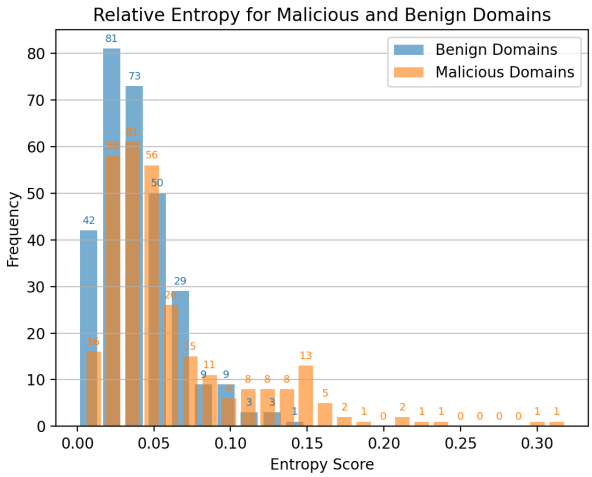

Figure 6: The Relative Entropy Score Frequency of 300 Malicious and 300 Benign Domains

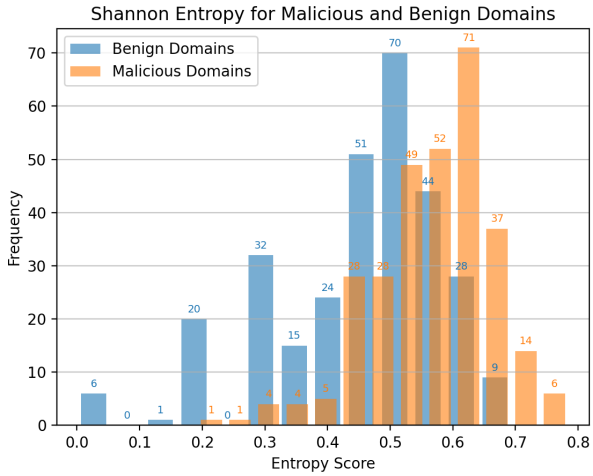

Figure 5: The Shannon Entropy Score Frequency of 300 Malicious and 300 Benign Domains

It is clear that malicious domains have a much shorter registration period than the benign domains. Although there are benign domains with low registration periods and malicious domains with high registration periods, they are uncommon. Thus, the domain registration period is also most likely a good feature.

3) Relative Entropy: In order to evaluate relative entropy, we have plotted two graphs. Figure 5 presents the frequency of Shannon entropy values of the benign and malicious domains. In Figure 6, the frequency of relative entropy values of malicious and benign domains is shown.

At first glance, the Shannon entropy seems to be the better choice for determining the randomness of a domain name. It can be seen in Figure 5 that the benign domains are distinguishable from the entropy score (x-axis) of 0.40 and lower. The malicious domains with random domain names generated by DGAs can be distinguished from an entropy score of 0.60 and higher, as shown in Figure 5. There are 128 malicious domains that had an entropy score of 0.60 or higher. A lot of the malicious domains that got a high Shannon entropy value (>= 0.60) are often long-named domains in English too, e.g., *connecthelpdesk.com*. Out of the 128 malicious domains, 22 (17%) domains were actually domains generated by a DGA. Thus, Shannon entropy is not ideal for detecting domain names generated by DGAs.

In comparison to the Shannon entropy graph in Figure 5, the relative entropy in Figure 6 shows that the majority of the benign domains and malicious domains have a score between 0.00 and 0.12. Specifically, the benign domains that are between the mentioned two values is 98%. There are 43 malicious domains with a relative entropy score of 0.12 or higher. Out of the 43 domains, 32 (74%) domains were generated by a DGA.

Comparing the Shannon entropy results with the relative entropy results, it can be seen that more randomized domain names of malicious domains are detected when relative entropy is used. In addition, there is less noise of non-random domains. Thus, relative entropy is more effective in detecting domains using randomized domain names generated by DGAs.

B. Subsystem Evaluation

1) Domain Classification System: After training the domain classification system with 80% of the training data, the classification system was able to correctly classify 96% of the control set. The results based on the evaluation dataset showed similar results, with an accuracy of 97%. The results of the evaluation dataset are shown in Table III. The model correctly classified 349 domains, of which 34 were correctly classified as malicious and 315 benign. The model also misclassified 10 domains, of which 2 domains were classified as malicious,

while it was benign. Furthermore, 8 domains were classified as benign, while it was malicious.

	Actually Positive	Actually Negative
Predicted Positive	34 (TP)	2 (FP)
Predicted Negative	8 (FN)	315 (TN)

Table III: Domain Classification Confusion Matrix

2) NetFlow Classification System: Disclosure: The NetFlow classification system correctly classified 91% of the control set after it had been trained. The result of the evaluation dataset, which can be seen in Table IV, shows that Disclosure misclassified 105 flows, from which 27 flows were classified as benign, while it was malicious. Looking at the FP, 78 flows were classified as malicious, while it was benign. Furthermore, Disclosure classified 354 domains correctly. The accuracy of Disclosure using the evaluation dataset is only 77%. However, we do have to emphasize that Disclosure is only able to detect 62% of all the malicious flows. Thus, the results show that the NetFlow classification system does not detect malicious flows well given our evaluation dataset.

	Actually Positive	Actually Negative
Predicted Positive	45 (TP)	78 (FP)
Predicted Negative	27 (FN)	309 (TN)

Table IV: NetFlow Classification Confusion Matrix

C. Proof of Concept Evaluation

The scores are combined by taking the accuracy of the two classification systems into consideration. The domain classification system had a higher accuracy than the NetFlow classification system. Therefore, we use a weighted mean to fairly combine two scores.

The PoC would give final verdict by making use of the following equation:

$$\bar{x} = \frac{x_1 w_1 + x_2 w_2}{w_1 + w_2} \tag{4}$$

where $\bar{x}$ is the final score, x_1 is the score given by the domain classification system, w_1 is the prediction accuracy of the domain classification system, x_2 is the score given by the NetFlow classification system, and w_2 is the accuracy of the NetFlow classification system.

After we completed the PoC system, we evaluated the system. The accuracy of our system was measured to be 81% on our evaluation dataset. The system correctly classified 67 out of 72 malicious flows and 308 out of 387 benign flows. Only 5 flows were misclassified as benign and 79 flows were falsely classified as malicious, these results are shown in Table V.

	Actually Positive	Actually Negative
Predicted Positive	67 (TP)	79 (FP)
Predicted Negative	5 (FN)	308 (TN)

Table V: Proof of Concept Classification Confusion Matrix

VII. Discussion

In this section, we summarize the results, highlight the importance of our findings, and discuss the shortcomings of our research.

A. Features Evaluation

We have evaluated the features used for our domain classification system to detect botnets. The selected features are the amount of NS, MX, and TXT RRs of a domain, the registration period of a domain, and the domain name entropy.

The evaluation of the features that were used for our domain classification system shows that the features that we have selected are effective in detecting malicious domains. The results of our re-evaluation are comparable with the paper that inspired us to use similar features [25].

The domain registration period is an effective feature for detecting malicious domains as the registration period is significantly shorter than benign domains. As for entropy, initially, it seems that Shannon entropy is more effective to use than relative entropy. However, our results show that relative entropy is more effective in detecting malicious domains that have randomized domain names that are generated by DGAs.

The result of this evaluation cannot only be used for this study, but it can also be used for future studies on detecting malicious domains using DNS.

The main shortcoming in this evaluation is the lack of using recent C&C botnet domain lists. We used lists that contain spam domains and we had to make an assumption that the attributes of these domains are similar to C&C botnet domains. This assumption could not be verified because there are no publicly disclosed recent C&C botnet domain lists. There is a list, named Spamhaus Block List, from the Spamhaus Project which contains the IP addresses and URLs associated spam sources and threats such as botnet C&C servers. However, this list is not publicly available.

A limitation of the domain registration period feature is that the information of WHOIS is not uniform. This means that the available information is not always the same for all the domains. We found instances in which the WHOIS result of the domain did not have information on the creation date, the expiration date, last updated date, or a combination of them. Thus, the calculation of the registration period had to be improvised, as mentioned in Section V-B1. Nevertheless, we think that results do show the general view, as the registration period of both benign and malicious domains were impacted.

Another limitation that we see in our research, is that we have only used Shannon entropy and relative entropy to detect randomness of a domain name. There are other methods in detecting randomness, such as n-gram analysis.

Another shortcoming in our study is that we have not re-evaluated the features used by Disclosure. This is something we had to accept for our study, due to the limited amount of time. Nevertheless, we do think it is of importance to evaluate whether the features used are effective for our PoC system.

B. Proof of Concept evaluation

The PoC was evaluated by measuring the accuracy of both our classification systems and the whole PoC system.

The evaluation of the domain classification system shows us that it was able to predict well given the training set and the evaluation dataset. Moreover, it shows that the selected features for this system were properly considered, which can be seen by the high accuracy. The false negative is relatively high, but our false positive was low. A solution to this problem is to lower the percentage at which a flow or domain is considered malicious. In our PoC this was set on 50% for both classification systems as described previously in Section IV.

The existing NetFlow classification system, named Disclosure, was not as accurate based on our experiment results. Although the accuracy of the control dataset was high, the evaluation dataset showed a significantly lower accuracy. In addition, the amount of detected malicious flows was low. The malware samples that we have used for our evaluation were not compatible with certain features of Disclosure. The evaluation set had malware samples which often only captured the initial malware traffic. Therefore, the evaluation set was not ideal for Disclosure. Moreover, it is known that Disclosure does not work well on malware that it has not been trained on.

The PoC system showed a 81% accuracy given the evaluation set. Additionally, our PoC was able to classify flows that do not have a domain associated with them. This allowed our PoC to detect more botnet traffic as opposed to when only the domain classification system is used. However, as the NetFlow classification system had a low accuracy, the PoC also had a lower accuracy than the domain classification system.

We have designed the final component of our PoC system based on the accuracy of the two classification systems. The result of the PoC shows that using two classification systems does not necessarily improve the accuracy in botnet detection. The PoC is able to overcome the limitation of a single classification system. It is able to use more data sources to classify the network traffic. However, the number of false positives had also increased. This significant increase is contributed by the limitations of the NetFlow classification system. A possible solution is to use a more accurate NetFlow classification system.

A shortcoming of evaluating the systems is that the evaluation dataset was relatively small. Unfortunately, there is no publicly known recent dataset available, that contains the DNS and NetFlow data of botnet traffic. CTU-13 is an example of such publicly available dataset, however, this dataset is not recent. The lifespan of a C&C domain is short, therefore the captured domains in the dataset would not be usable anymore for our system. Our improvised dataset contained samples of botnet traffic. This is not ideal as the NetFlow classification system makes use of temporal features, which might not be present in all datasets. Another limitation of our improvised dataset is the size. Due to the limited amount of NetFlow flows and domains, we were unable to accurately measure the performance of our system. However, we were able to show that our botnet detection approach is promising.

VIII. Conclusion

In this research, we looked at ways of detecting botnet traffic to transient C&C servers using NetFlow and DNS data. We looked at the characteristics C&C domains have and how they differ from benign domains. We found that the amount of DNS resource records, relative domain entropy, and the registration period differ between C&C and benign domains. A domain classification system was created using these features. Additionally, we looked at NetFlow characteristics of botnet and benign traffic. However, we chose to use an existing botnet detector using NetFlow data, named Disclosure. This botnet detector makes use of flow sizes, access patterns, and temporal behavior as NetFlow features.

We found that domain classification could achieve high true positive rates while having minimal false positives and negatives by making use of the number of resource records, the domain registration period, and domain entropy. In our experiments, we measured a 97% accuracy rate given the evaluation dataset. However, due to the limited size of the evaluation dataset, we cannot conclusively say whether the same accuracy rates can be expected on other datasets.

Using the NetFlow classification system we measured an accuracy rate of 77% based on the evaluation dataset. Due to the difference in accuracy, we chose to use a weighted mean to combine the two scores of the two systems. By combining the two systems we were able to achieve a 81% accuracy rate on the evaluation dataset. Moreover, the PoC overcomes the limitation of a single classification system, which allows it to use more data sources to classify network traffic. Although there are some limitations in our evaluation dataset, we can conclude that combining DNS and NetFlow data can be an effective way to detect botnet C&C servers. However, a better NetFlow classification system should be considered.

IX. Future Work

Our research has shown how NetFlow and DNS data can be combined to detect traffic to transient command and control servers. However, our PoC system has been evaluated using limited, publicly available datasets. Future research could evaluate the system using more extensive real-world datasets, which will likely lead to more accurate results than we could achieve. Additionally, in this research, we have evaluated the features used in the domain classification system. Future research should evaluate the features used in the NetFlow classification system, as the systems' prediction accuracy relies on the quality of the features used.

This research has shown remarkable results using DNS classification, however, NetFlow classification did not perform well. Future research could review the currently used NetFlow features and experiment using other features to improve the prediction accuracy of the NetFlow classification system. Improving the NetFlow classification system would lead to better detection results and fewer false positives and negatives.

The classification systems could also benefit from other features. Future research could experiment using other features such as the registrar, the contents of DNS records, where a domain is registered, the BGP AS number, and where an IP is located. More features could lead to a better detection rate as well as making the system more reliable. However, this should also be evaluated.

References

[1] Leyla Bilge et al. "Disclosure: Detecting Botnet Command and Control Servers through Large-Scale NetFlow Analysis". In: *Proceedings of the 28th Annual Computer Security Applications Conference*. ACSAC '12. Orlando, Florida, USA: Association for Computing Machinery, 2012, pp. 129–138. ISBN: 9781450313124. DOI: 10.1145/2420950.2420969. URL: https://doi.org/10.1145/2420950.2420969.

[2] The Spamhaus Project. *Botnet Threat Report 2019*. URL: https://www.spamhaus.org/news/images/full-2019/spamhaus-botnet-threat-report-2019.pdf (visited on June 15, 2020).

[3] Malwarebytes Labs. *2020 State of Malware Report*. URL: https://resources.malwarebytes.com/files/2020/02/2020_State-of-Malware-Report.pdf (visited on June 15, 2020).

[4] Yury Zhauniarovich et al. "A Survey on Malicious Domains Detection through DNS Data Analysis". In: *ACM Comput. Surv.* 51.4 (July 2018). ISSN: 0360-0300. DOI: 10.1145/3191329. URL: https://doi.org/10.1145/3191329.

[5] Igor Mishsky, Nurit Gal-Oz, and Ehud Gudes. "A Topology Based Flow Model for Computing Domain Reputation". In: *Data and Applications Security and Privacy XXIX*. Ed. by Pierangela Samarati. Cham: Springer International Publishing, 2015, pp. 277–292. ISBN: 978-3-319-20810-7. DOI: 10.1007/978-3-319-20810-7_20. URL: https://doi.org/10.1007/978-3-319-20810-7_20.

[6] Manos Antonakakis et al. "Building a dynamic reputation system for dns." In: *USENIX security symposium*. 2010, pp. 273–290. URL: https://www.usenix.org/legacy/event/sec10/tech/full_papers/Antonakakis.pdf (visited on June 5, 2020).

[7] Jérôme François et al. "BotTrack: Tracking Botnets Using NetFlow and PageRank". In: *NETWORKING 2011*. Ed. by Jordi Domingo-Pascual et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1–14. ISBN: 978-3-642-20757-0. DOI: 10.1007/978-3-642-20757-0_1. URL: https://doi.org/10.1007/978-3-642-20757-0_1.

[8] G. Vormayr, T. Zseby, and J. Fabini. "Botnet Communication Patterns". In: *IEEE Communications Surveys Tutorials* 19.4 (September 2017), pp. 2768–2796. ISSN: 1553-877X. DOI: 10.1109/COMST.2017.2749442. URL: https://doi.org/10.1109/COMST.2017.2749442.

[9] Zonghua Zhang, Ruo Ando, and Youki Kadobayashi. "Hardening Botnet by a Rational Botmaster". In: *Information Security and Cryptology*. Ed. by Moti Yung, Peng Liu, and Dongdai Lin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 348–369. ISBN: 978-3-642-01440-6. DOI: 10.1007/978-3-642-01440-6_27. URL: https://doi.org/10.1007/978-3-642-01440-6_27.

[10] I. Ghafir and V. Prenosil. "Blacklist-based malicious IP traffic detection". In: *2015 Global Conference on Communication Technologies (GCCT)*. April 2015, pp. 229–233. DOI: 10.1109/GCCT.2015.7342657.

[11] Paul Mockapetris. *Domain names: Concepts and facilities*. RFC 882. November 1983. DOI: 10.17487/RFC0882. URL: https://rfc-editor.org/rfc/rfc882.txt.

[12] Paul Mockapetris. *Domain names: Implementation specification*. RFC 883. November 1983. DOI: 10.17487/RFC0883. URL: https://rfc-editor.org/rfc/rfc883.txt.

[13] Paul Mockapetris. *Domain names - concepts and facilities*. RFC 1034. November 1987. DOI: 10.17487/RFC1034. URL: https://rfc-editor.org/rfc/rfc1034.txt.

[14] Paul Mockapetris. *Domain names - implementation and specification*. RFC 1035. November 1987. DOI: 10.17487/RFC1035. URL: https://rfc-editor.org/rfc/rfc1035.txt.

[15] Ken Harrenstien and Vic White. *NICNAME/WHOIS*. RFC 812. March 1982. DOI: 10.17487/RFC0812. URL: https://rfc-editor.org/rfc/rfc812.txt.

[16] Leslie Daigle. *WHOIS Protocol Specification*. RFC 3912. September 2004. DOI: 10.17487/RFC3912. URL: https://rfc-editor.org/rfc/rfc3912.txt.

[17] Ken Harrenstien, Mary Stahl, and Elizabeth Feinler. *NICNAME/WHOIS*. RFC 954. October 1985. DOI: 10.17487/RFC0954. URL: https://rfc-editor.org/rfc/rfc954.txt.

[18] Benoît Claise. *Cisco Systems NetFlow Services Export Version 9*. RFC 3954. October 2004. DOI: 10.17487/RFC3954. URL: https://rfc-editor.org/rfc/rfc3954.txt.

[19] Daniel Plohmann et al. "A Comprehensive Measurement Study of Domain Generating Malware". In: *25th USENIX Security Symposium (USENIX Security 16)*. Austin, TX: USENIX Association, August 2016, pp. 263–278. ISBN: 978-1-931971-32-4. URL: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/plohmann (visited on July 1, 2020).

[20] C. E. Shannon. "A mathematical theory of communication". In: *The Bell System Technical Journal* 27.3 (July 1948), pp. 379–423. ISSN: 0005-8580. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[21] S. Kullback and R. A. Leibler. "On Information and Sufficiency". In: *The Annals of Mathematical Statistics* 22.1 (1951), pp. 79–86. ISSN: 00034851. URL: http://www.jstor.org/stable/2236703 (visited on July 2, 2020).

[22] Michael E Tipping. "Sparse Bayesian learning and the relevance vector machine". In: *Journal of machine learning research* 1.Jun (2001), pp. 211–244. URL: http://www.jmlr.org/papers/volume1/tipping01a/tipping01a.pdf (visited on July 4, 2020).

[23] Jonghoon Kwon et al. "PsyBoG: A scalable botnet detection method for large-scale DNS traffic". In: *Computer Networks* 97 (2016), pp. 48–73. ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2015.12.008. URL: http://www.sciencedirect.com/science/article/pii/S1389128615004843.

[24] S.H. Mousavi, M. Khansari, and R. Rahmani. "A fully scalable big data framework for Botnet detection based on network traffic analysis". In: *Information Sciences* 512 (2020), pp. 629–640. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2019.10.018. URL: http://www.sciencedirect.com/science/article/pii/S0020025519309703.

[25] Masahiro Kuyama, Yoshio Kakizaki, and Ryoichi Sasaki. "Method for detecting a malicious domain by using whois and dns features". In: *The third international conference on digital security and forensics (DigitalSec2016)*. Vol. 74. 2016. URL: https://www.researchgate.net/profile/Natalie_Walker4/publication/307605969_Proceedings_of_the_Third_International_Conference_on_Digital_Security_and_Forensics_DigitalSec_Kuala_Lumpur_Malaysia_2016/links/57cd191508ae89cd1e86d434/Proceedings-of-the-Third-International-Conference-on-Digital-Security-and-Forensics-DigitalSec-Kuala-Lumpur-Malaysia-2016.pdf#page=76 (visited on June 13, 2020).

[26] Nizar Kheir et al. "Mentor: Positive DNS Reputation to Skim-Off Benign Domains in Botnet C&C Blacklists". In: *ICT Systems Security and Privacy Protection*. Ed. by Nora Cuppens-Boulahia et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 1–14. ISBN: 978-3-642-55415-5. DOI: 10.1007/978-3-642-55415-5_1. URL: https://doi.org/10.1007/978-3-642-55415-5_1.

[27] DomainTools. *The Distribution of Malicious Domains*. URL: https://www.domaintools.com/content/The_DomainTools_Report_Distribution_Malicious_Domain.pdf (visited on July 10, 2020).

[28] *Malware-Traffic-Analysis.net - My technical blog posts - 2020*. URL: https://www.malware-traffic-analysis.net/2020/index.html (visited on June 15, 2020).

[29] *The CTU-13 Dataset. A Labeled Dataset with Botnet, Normal and Background traffic. — Stratosphere IPS*. URL: https://www.stratosphereips.org/datasets-ctu13 (visited on June 15, 2020).

APPENDIX A
PROOF OF CONCEPT GIT REPOSITORY

We have created a Git repository for this study. In this Git repository, one is able to find all the files that were used to perform this study. The repository can be found at: https://github.com/os3-rp2-2020/RP2