A Framework for DNS Based Detection and Mitigation of Malware Infections on a Network

Etienne Stalmans[1] and Barry Irwin[2]
Security and Networks Research Group
Department of Computer Science
Rhodes University
Grahamstown,South Africa
E-mail: g07s0924@campus.ru.ac.za[1]; b.irwin@ru.ac.za[2]

***Abstract*—Modern botnet trends have lead to the use of IP and domain fast-fluxing to avoid detection and increase resilience. These techniques bypass traditional detection systems such as blacklists and intrusion detection systems. The Domain Name Service (DNS) is one of the most prevalent protocols on modern networks and is essential for the correct operation of many network activities, including botnet activity. For this reason DNS forms the ideal candidate for monitoring, detecting and mitigating botnet activity. In this paper a system placed at the network edge is developed with the capability to detect fast-flux domains using DNS queries. Multiple domain features were examined to determine which would be most effective in the classification of domains. This is achieved using a C5.0 decision tree classifier and Bayesian statistics, with positive samples being labeled as potentially malicious and negative samples as legitimate domains. The system detects malicious domain names with a high degree of accuracy, minimising the need for blacklists. Statistical methods, namely Naive Bayesian, Bayesian, Total Variation distance and Probability distribution are applied to detect malicious domain names. The detection techniques are tested against sample traffic and it is shown that malicious traffic can be detected with low false positive rates.**

I. Introduction

Modern corporate systems provide multiple potential vectors for infection by malicious software known as malware [1]. Once a system has been infected, the malware can be used for data stealing, sending spam, phishing, distributing malware and distributed denial of service attacks. Current detection methods rely on host-based malware detection that are based on pattern matching and heuristics. These traditional detection techniques are easily bypassed by zero-day attacks and polymorphic code. Current network based solutions often focus on preventing malware from entering the system through the use of firewalls, Intrusion Detection Systems and blacklists. These systems are blind to malware that enters the system through other attack vectors such as mobile internet connections and removable devices. The current generation of malware is largely focused on the creation of large networks of infected hosts, known as botnets.

Botnets consist of thousands of infected hosts which receive instructions from command and control (C&C) servers operated by an individual. Traditionally IRC servers have been used as C&C servers and communicated with the botnet through IRC channels. This lead to network administrators often blocking IRC traffic on the network. Recent trends in botnet development have seen the use of alternative communication channels, such as DNS-tunnelling and HTTP, between the C&C servers and infected hosts [2]. The use of alternative communication channels have allowed botnets to bypass common network filters [3]. Furthermore, these channels cannot be blocked as simply as IRC traffic has been, as they are essential for normal network activity. Furthermore, recent botnets such as Conficker, Kraken and Torpig have used Domain Name Server (DNS) fast-flux to avoid detection and to reduce the ability of researchers to find and shut-down the C&C servers. A new method based on fast-flux has emerged where each bot algorithmically generates a large set of domain names to query [4]. Trends in algorithmic name generation has seen bots, such as those infected with Conficker-C, generating upwards of 50 thousand domain names an hour [5]. This makes it near impossible for researchers to block or pre-register all domains associated with these algorithmically generated domains, as was done with earlier Conficker variants [5], [6]. Furthermore, the large volume of generated names makes the maintaining and using of domain blacklists slow and cumbersome.

As such, a novel system for monitoring DNS traffic at the network egress points is proposed. Egress filtering will allow for automatic detection and containment of malware activity. Through the use of Bayesian modelling of DNS traffic similarity, it is possible to detect infected hosts on a network. The identification of irregular domain names and recording of unresolved DNS queries can be used in the detection of DNS fast-flux queries and subsequently command and control servers, as well as infected hosts on the internal network.

This paper discusses related work in Section II, the datasets used are described in Section III. The DNS and domain name features examined are explained in Section IV, while the classification models used are detailed and discussed in Section V. The results obtained from testing sample traffic is presented in Section VI. These results are discussed in Section VII, with concluding remarks in Section VIII.

II. Related Work

A number of approaches for detecting malicious network activity through DNS traffic monitoring were studied. Perdisci,

Corona, Dagon and Lee implemented a system for the detection of malicious fast-flux service networks through the passive analysis of recursive DNS traffic traces [7]. Common features in malicious fast-flux DNS traffic were identified, such as a short time-to-live (TTL) and multiple Address (A) records. IP addresses resolved to the domain name were often from dissociated networks and changed rapidly.

The work of Holz, Gorecki, Reck, Freiling and Felix [8], identified the same key domain features. These were used for the creation of heuristic classification models for the detection of fast-flux botnets. The primary observation was that fast-flux botnets could be detected using the number of distinct A records returned and the number of different Autonomous System Number (ASN) [9] associated with the domain. Results showed botnet creators mimicking the structure of Content Distribution Networks (CDN). This behaviour masks botnet activity and hinders the automatic classification of domains [8]. Holz *et al*. identified the inherent distributed structure of botnets as a distinguishing factor. The A record counts, along with the number of distinct ASNs, provide accurate identification of fast-flux botnets [8]. It was further noted that these features were not easily obfuscated by botnet controllers. The detection system proposed required secondary DNS queries once the original queries TTL expired. This increases the time required to classify domains and may not be viable in larger networks. Attempts at outbound malware traffic detection have primarily relied on host-based detection [10]. Zang, Perdisci, Gu and Lee did however implement a system for botnet detection at the network edge [11], [12]. The proposed system placed sensors at the network edge, increasing the volume of traffic each sensor monitored, as opposed to previous host-based systems. The increased traffic volumes allowed for the correlation of network activity over time.

Reddy A.L, Reddy A.K, Yadav and Ranjan proposed a system, based on signal theory, for detecting algorithmically generated domain names [4]. They looked at the distribution of alphanumeric characters as well as the distribution of bi-grams within domain names. This technique required the evaluation of domain names mapping to the same set of IP addresses. At least 50 domain names mapping to the same set of IP-addresses were required to positively identify malicious domains. This technique could allow malicious traffic to egress from the network before a system has been able to classify a domain as possibly malicious.

This paper proposes alternative classification metrics, allowing for accurate classification of algorithmically generated domain names from a single domain query. URLs used for phishing and advertising spam were analysed by Ma, Saul, Savage and Voelker [13]. They identified that malicious URLs exhibit different alphanumeric distributions than legitimate URLs. Statistical learning techniques were employed to identify malicious URLs from lexical features such as domain name length, number of dots in the URL and host names. The proposed system aimed to identify single URLs as malicious, whereas Reddy *et al.* required URL grouping for accurate classification [4], [13]. Work performed by Xie, Yu, Achan, Panigrahy, Hulten and Osipkov lead to the development of regular expression based signatures for detection of spam URLs [14]. The solution proposed in this paper is intended to surpass the accuracy of this regular expression based solution. Furthermore, the solution should be harder to bypass and will avoid the need to constantly update signatures to match new attacks as they develop. The use of statistics in detecting malicious activity has been included in numerous studies, including botnet detection [15], email spam-filtering [16] and intrusion detection [17].

III. Datasets

The datasets used were divided into *training* data and *test* data. The division of datasets was done to ensure that the classifiers were adequately trained to recognise malicious domains but that the results would not be tainted by using known data during testing.

The data contained in the training dataset was manually verified and labeled as malicious or legitimate. The legitimate domain data was obtained from the Google Doubleclick ad planner top-1000 most visited sites list [18]. The malicious data was taken from multiple sources, including the fast-flux trackers for ZeuS [19], SpyEye [20] and other botnets [21]. The data used for training the classifiers for algorithmically generated domain names consisted of samples taken from domain names generated by Kraken [22] and Conficker-C [6].

Testing was performed using DNS traffic logged at a large South African University and a local schools network. The datasets consisted of a .pcap dump containing 40,910,498 raw DNS packets. And a secondary dump of 33,261,575 visited URLs along with timestamps as seen by the Squid proxy. A secondary set of test data was obtained from MalwareURL [23], listing 137,747 malicious URLs.

IV. Features

The central goal of this research is to classify a domain as either potentially malicious or legitimate (non-malicious) with a high degree of reliability. The domain classification problem is treated as a binary classification problem, where positive samples will be labeled as malicious domains and negative samples as legitimate domains. Multiple domain features were examined to determine which would be most effective in the classification of domains.

A. DNS Features

DNS query responses were examined to identify key features that would be useful in the classification of domains. The features selected were present in both legitimate and malicious domain queries. Table I lists the common DNS query response features used for classification, along with the average values they contain for standard domains, content distribution networks (CDNs) and fast-flux domains based on results obtained from the observation of legitimate and malicious training data. Fast-flux domains share many common characteristics with CDNs, making accurate classification more difficult. The training data shows that fast-flux domains have the shortest average

Table I
FEATURES USED FOR DOMAIN CLASSIFICATION, ALONG WITH AVERAGE VALUES FOR MULTIPLE DOMAIN TYPES

	Standard DNS	CDN	Fast-Flux
A Records	4	4	4
NS Records	2	2	2
Network Ranges	1	1	3
Unique ASNs	1	1	2
TTL	≥ 1800	< 1800	≤ 600

TTL of all observed domain types. Furthermore, the domain hosts are spread across multiple, widely dispersed IP ranges, typically more than three. The domain hosts are associated with multiple Autonomous System Numbers (ASNs), typically two or more. CDNs display similar characteristics, but on average are associated with fewer ASNs and resolve to less widely dispersed IP ranges. The nature of fast-flux domains dictates that they are associated with recently registered domains. This feature could be used to distinguish between legitimate CDNs and fast-flux domains. The registrar information may be obtained through a *whois* query. Information returned by a *whois* query includes the registration date, the registration authority and the country the domain was registered in. Table II shows an example DNS query response for an active fast-flux domain, where the C&C servers are widely dispersed over multiple ASNs, countries and net-blocks.

Table II
SAMPLE FAST-FLUX DNS QUERY RESPONSE

champiogogo.ru				
IP Address	Net block	ASN	Country	TTL
60.13.74.23	60.13.64.0/18	4837	CN	300
62.42.100.212	62.42.0.0/16	6739	ES	300
148.217.94.55	148.217.0.0/16	6503	MX	300
212.69.189.125	212.69.160.0/19	8218	DE	300
217.217.199.129	217.216.0.0/15	6739	ES	300

B. Textual Features

Domain names provide a readable and easy to remember mapping between a domain and its address, typically consisting of English words or a combination of English words. Malicious domains, which are algorithmically generated, such as those associated with Conficker (*xllnm.com.do*) and Kraken (*ygcoqgmmb.yi.org*) commonly do not contain English words or letter combinations usually seen in legitimate domain names. As seen in Figure 1, it is possible to calculate letter frequencies as they occur in legitimate, randomly generated and algorithmically generated domain names. Statistical methods are applied to these frequency distributions, allowing for accurate classification of domain names as legitimate or malicious, based solely on alphanumeric character distribution.

V. CLASSIFICATION MODELS

A. Fast-flux Detection

Several different classification models for the identification of fast-flux domain queries were evaluated. Initially techniques

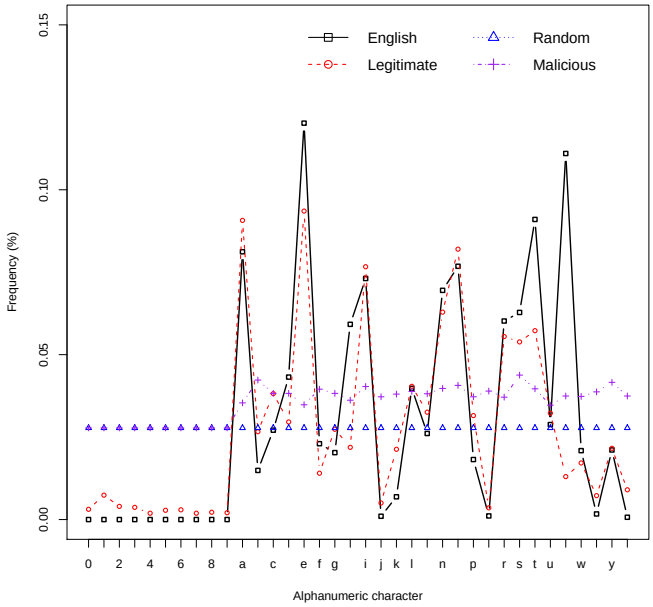


Figure 1. Frequency graph for alphanumeric letter distribution in domain names

similar to the work by Holz *et al.* were employed, with the DNS features identified by Holz *et al.* being used in an expert system to detect fast-flux domain queries [8]. Analysis of the approach employed by Holz *et al.* revealed a delay in domain classification, as a secondary DNS query had to be performed once the original queries TTL had expired. To overcome this delay two different classifiers were employed with the aim of classifying a domain as fast-flux from a single domain query. The first classifier employed was the C5.0 [24] decision-tree classifier. While the second classifier was based on a Naive Bayesian classifier, applying statistical knowledge of previous fast-flux domains.

1) C5.0 Decision-tree Classifier: A C5.0 decision-tree classifier was used to automatically classify a DNS query response as either fast-flux or legitimate [24]. The decision-tree was constructed using training data, consisting of test domains which had been manually evaluated and labeled as either fast-flux or legitimate. Using the C5.0 decision-tree classifier has several benefits, the decision-tree constructed from the training data, produces readable and easy to interpret decision statements. These can be examined to identify features that are most likely to identify fast-flux domains. Furthermore, the C5.0 classifier produces accurate results and increases in accuracy when exposed to larger training data sets.

2) Bayesian Classifier: Bayesian inference is a statistical technique that is useful in the classification of problem domains which have a binary outcome. The aim was to classify a domain as either fast-flux or legitimate. Using a Bayesian classifier token, such as TTL, unique ASNs and A record counts, were correlated to calculate the probability of a domain

being fast-flux or not. Bayesian classifiers have been used to solve other computer security related problems, particularly email spam-filtering. In spam-filtering words found in emails are given probabilities of occurring in spam email and legitimate email [16]. Probabilities are calculated from training data that had been manually classified as either spam or legitimate. Once a classifier had been trained it could be used for classification. The classifiers outputs a likelihood total, which could be used to mark a domain as fast-flux or not [25]. The classifier will use a formula derived from Bayes' theorem to classify DNS query responses.

$$P(F \mid t) = \frac{P(t \mid F).P(F)}{P(t \mid F).P(F) + P(t \mid \neg F).P(\neg F)}$$

Where:

- $P(F \mid t)$ is the probability that a domain is fast-flux, if the token *t* is in the response;
- $P(F)$is the overall probability that a domain is fast-flux;
- $P(t \mid F)$ is the probability that the token appears in a fast-flux domain query responses;
- $P(\neg F)$ is the overall probability that a domain is legitimate (not fast-flux);
- $P(t \mid \neg F)$ is the probability that the token appears in a legitimate domain query responses.

3) Naive Bayesian Classifier: The Bayesian classifier used the overall probability that a domain was fast-flux (P(F)) or legitimate ($P(\neg F)$). Legitimate domains could be incorrectly labeled as fast-flux if the system was exposed to a high number of fast-flux domain queries over a period of time. The bias could be eliminated by using a Naive Bayesian classifier. The Naive Bayesian classifier assumes no overall probability that a domain is fast-flux or not. The classifier is expressed using the following formula, with the assumption that $P(F)$ and $P(\neg F)$ both equal 0.5.

$$ln\frac{P(F \mid D)}{P(\neg F \mid D)} = ln\frac{P(F)}{P(\neg F)} + \sum_i ln\frac{P(t_i \mid F)}{P(t_i \mid \neg F)}$$

Where:

- $ln\frac{P(F|D)}{P(\neg F|D)}$ is the logarithmic probability ratio that a domain is fast-flux ($P(F \mid D)$) or legitimate ($P(\neg F \mid D)$)
- $P(F)$is the overall probability that a domain is fast-flux;
- $P(\neg F)$ is the overall probability that a domain is legitimate (not fast-flux);
- $P(t_i \mid F)$ is the probability that the token appears in a fast-flux domain query responses;
- $P(t_i \mid \neg F)$ is the probability that the token appears in a legitimate domain query responses.

The output of the Naive Bayesian classifier will serve as input for the Bayesian classifier, determining the weighting of the overall probabilities $P(F)$ and $P(\neg F)$.

B. Algorithmically generated domain name detection

The second part of the system used the textual features of domain names to detect algorithmically generated domain names. The values of the domain labels are examined, while ignoring the top-level domain values. The system used similar detection metrics as used for the fast-flux detection. The Naive Bayesian and Bayesian classifiers are the same as discussed in Section V-A2 and Section V-A3, where the alphanumeric characters present in the domain name are used as tokens. Additional statistical methods were applied to aid in more accurately classifying domain names as algorithmically generated or not.

1) Probability distribution: During the training period, the probabilities of alphanumeric characters occurring in legitimate and algorithmically generated domain names were calculated. Figure 1 graphs the frequency values obtained during training. The large difference in probabilities of occurrence in the two datasets made it possible to determine whether a domain has been algorithmically generated by calculating the product of the probabilities as they occur in the two different datasets. The domain name is labeled as legitimate or not based on the larger probability product. Total variation distance is used to formally define the difference between the two probability distributions and is discussed in Section V-B2.

2) Total variation distance: In statistics the total variation distance is the maximum possible difference between two probability distributions that can be assigned to a single event. The variation distance was used to gauge the difference between the probability of a domain name being algorithmically generated or legitimate. The total variation distance is formally defined as:

$$\sigma(P, Q) = \frac{1}{2}\sum_i \mid P(x) - Q(x) \mid$$

Where:

- $P(x)$is the probability of x occurring in a legitimate domain name;
- $Q(x)$is the probability of x occurring in an algorithmically generated domain name.

VI. Results

A. Fast-flux detection

The aim was to classify a domain as either fast-flux or not from a single DNS query response, as opposed to previous work by Holz *et al.* which required a second DNS query once the TTL of the original query had expired [8]. Reducing the amount of time required to classify a domain, while maintaining high accuracy was essential. Furthermore, any interaction with the suspect domain was minimised. Multiple classification techniques were used to meet the aim of increasing the accuracy of fast-flux domain detection.

1) C5.0 classifier: C5.0 is used in data mining to extract patterns from databases, which can then be expressed as decision trees. The C5.0 provided by Rulequest Research was used to construct a decision tree from the training data. From the C5.0 classifier the DNS features most likely to indicate a fast-flux domain were extracted. Key features identified were: the number of different network ranges, the total number of

Table III
MEAN VALUES OF DNS FEATURES FOR FAST-FLUX AND LEGITIMATE DOMAINS

	A Records	NS Records	Number IP ranges	Number ASNs	TTL
Fast-Flux	2.090032	3.916399	2.180064	3.70418	594.9968
Legitimate	1.730769	3.87574	0.1538462	1.094675	14885.42

A records and the number of different ASNs. These features were then used to construct the classifier. The output of the classifier was combined with that of the other classifiers to give an overall flux-score.

2) Naive Bayesian classifier: The Naive Bayesian classifier assumes that a domain has equal initial probabilities of being fast-flux or legitimate. The different features of the DNS query response are then used to determine the fast-flux/legitimate domain likelihood ratio. Table III shows the means of different DNS features obtained from the Start of Authority (SOA) records in the test data. The means are vastly different for fast-flux and legitimate domain queries, allowing for the accurate classification of domains. Table IV shows the output of the Naive Bayesian classifier. The results clearly show that both fast-flux and CDN domains have been correctly classified. Both *wordpress.com* and *yahoo.com* are correctly identified as legitimate CDN, despite the query response closely resembling a fast-flux domain.

Table IV
OUTPUT OF NAIVE BAYESIAN CLASSIFIER FOR FAST-FLUX AND CDN DOMAIN QUERIES

Domain	Safe Score	Malicious Score	Classification
gingerbucksea.com	0.005304578	0.3550235	Fast-flux
pearlrumor.ru	3.059976e-14	7.490562e-13	Fast-flux
wordpress.com	1.536894e-08	4.250896e-10	Legitimate
championogogo.ru	3.395984e-09	1.723838e-06	Fast-flux
yahoo.com	1.940412e-15	1.509179e-69	Legitimate

B. Algorithmically generated domain name detection

The classifiers aim to identify malicious domain names. For a fully qualified domain name (FQDN), such as *login.mysite.com*, *mysite* is referred to as the second-level domain and *com* as the first-level domain. The third-level domain, *login*, is what was examined and is henceforth referred to as the *domain name*. Multiple detection metrics were examined to determine which would yield the most accurate results. A high true positive rate with a minimal fast positive rate was desired. It was determined that a lower false positive rate was favourable, as fewer domains would be incorrectly labeled as malicious., leading to minimal downtime and disruption in network traffic. Table V shows the accuracy, true positive and the false positive rates for the four classifiers examined. The results were obtained after training each classifier with the same dataset, consisting of a 1000 samples of legitimate domains and a 1000 algorithmically generated domains. The algorithmically generated domain names were taken from both Kraken and Conficker-C samples [5], [22]. In general, it was

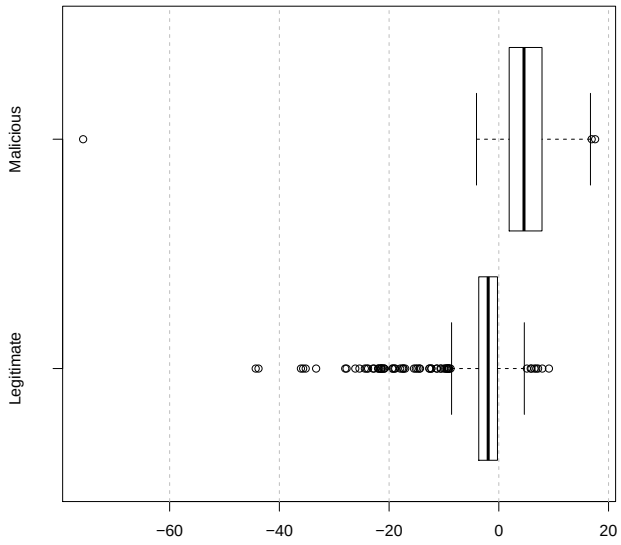


Figure 2. Box plot showing density distribution of Naive Bayesian classifier output for malicious and legitimate domain names

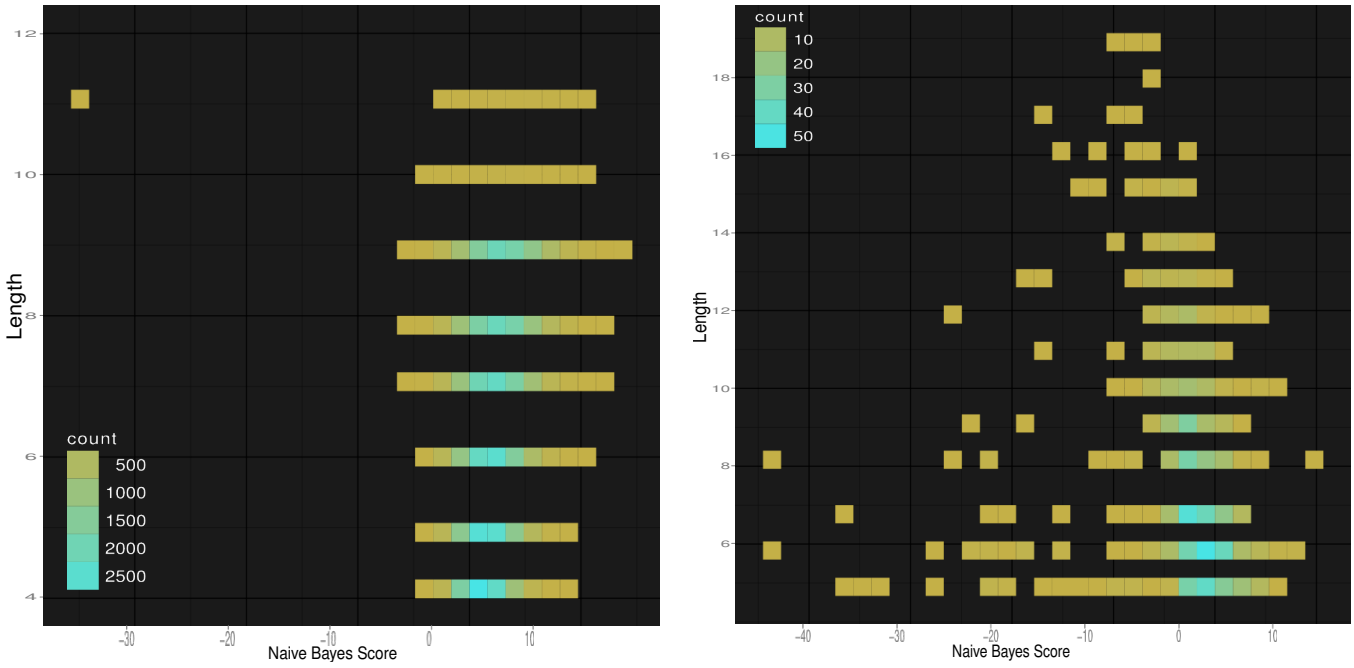
observed that classifiers based on Bayesian theory provided higher accuracy rates, with accuracy increasing when exposed to larger training datasets.

Table V
RESULTS OF THE FOUR CLASSIFIERS

Classifier	Accuracy	TPR	FPR
Naive Bayesian	87%	82%	8%
Variation	82%	80%	17%
Probability	84%	86%	17%
Bayesian	85%	81%	11%

TPR:True Positive Rate
FPR:False Positive Rate

1) Naive Bayesian classifier: The likelihood ratio was used to classify the domain name as malicious or legitimate, based on the distribution of alphanumeric characters. The Naive Bayesian formula outlined in Section V-A3 was used to compute the likelihood ratio. The density distribution of the classifier output for malicious and non-malicious domains has been visualised using the box-plot seen in Figure 2. The box-plot confirms the distribution pattern shown in the heat map plots, in Figures 3. The output for legitimate domains is grouped below 0.5, with a few outliers greater than 0.5



(a) Malicious domains

(b) Legitimate domains

Figure 3. Heat Maps of results from Naive Bayes classifier for Malicious and Legitimate domains

as well as multiple outliers much less than 0.5. The density distribution for malicious output clearly lies about 0.5 with very few outliers on either side of 0.5.

Figure 3(b) shows the output from the classifier when exposed to legitimate domains. It is possible to observe that for legitimate domains, the classifiers output is less than 0.5 for the vast majority of legitimate domains. The average value of the output decreases as the domain name length increases. Figure 3(a) shows the output from the classifier when exposed to malicious domains. As opposed to the results seen in Figure 3(b), the output is extensively greater than 0.5. Furthermore, as indicated by the heat-map, the number of domains that produce an output far larger than 0.5 increases with domain name length. This, combined with the results shown in Figure 3(b), indicates that the accuracy of the classifier will improve with longer domain names.

Table VI
DOMAIN NAMES AND THE OUTPUT AS PRODUCED BY THE NAIVE BAYES CLASSIFIER

Domain name	Output	Classification	Correct
Facebook.com	-1.06400	Legitimate	Yes
allrecipes.com	-4.25654	Legitimate	Yes
twitter.com	-2.39181	Legitimate	Yes
buzzle.com	2.47540	Malicious	No
nhk.or.jp	0.64375	Malicious	No
bbhkxkjh.com.fj	6.61512	Malicious	Yes
pveufjtm.com.bo	3.25285	Malicious	Yes
rrxwigqj.am	5.24226	Malicious	Yes
ljtkrinq.com.tt	2.75078	Malicious	Yes

Table VI shows output from the classifier for a few sample domains. The domains that have been classified correctly as malicious and legitimate, would easily be identified by a human observer. However the false positives, where legitimate domains have been classified as malicious, are more ambiguous. These domain names do not look like standard English. In the case of *nhk.or.jp* being identified, it is clear that this occurred due to the short length of the domain name as well as the combination of characters. This character combination does not usually appear together in legitimate or English words. The domain *buzzle.com* has a high malicious score, and is attributed to the double appearance of z which has an extremely low probability of occurring in a legitimate domain (0.009), but a high probability of occurring in a malicious domain (0.04), as shown in Figure 1.

2) Bayesian classifier: The Naive Bayesian classifier provides accurate identification of algorithmically generated domain names without assuming a prior bias, Using the result of the Naive Bayesian classifier as the bias for the standard Bayesian classifier, a higher detection accuracy was achieved when compared to a standard Bayesian classifier. However, as seen in Table V, the accuracy of the Bayesian classifier is lower than that of the Naive Bayesian classifier, as well having a higher false positive rate. The true positive rate of the Bayesian classifier is similar to that of the Naive Bayesian classifier.

3) Total Variation distance classifier: Figure 4 shows the results of the total variation distance classifier. There is a clear overlapping of classification scores between algorithmically generated (malicious) and legitimate domains, especially for

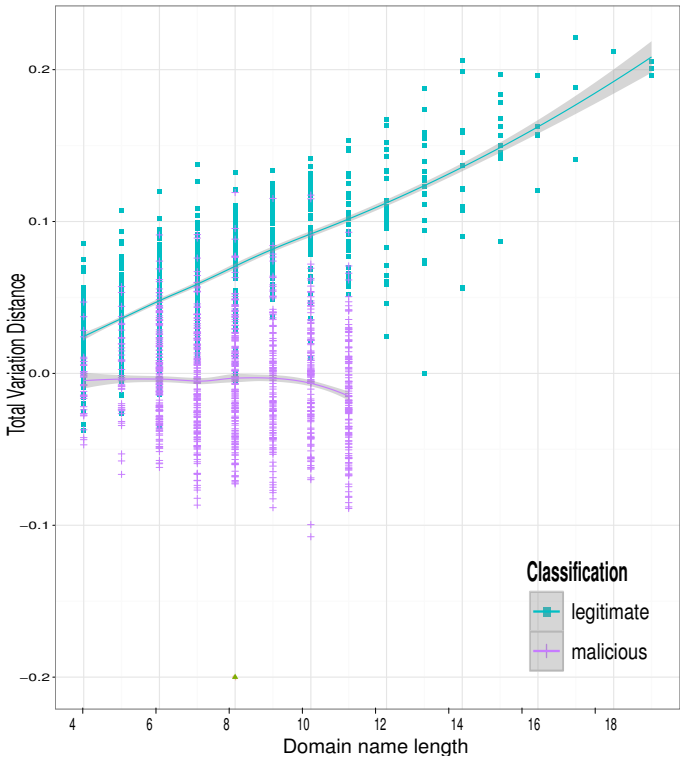


Figure 4. Scatter plot showing the total variation distance for malicious and legitimate domain names.

shorter domain names. The overlapping decreases as domain name length increases and total variation distance decreases for malicious domain names, while increasing for non-malicious domain names. The spline smoothed lines clearly show how total variation distance increases for legitimate domain names when the domain name length increases. The results show a 82% accuracy for the total variation distance classifier, although a far higher false positive rate exists than in the Bayesian based classifiers.

4) Probability classifier: Results obtained by the probability classifier show similar trends to that of the total variation distance classifier, expressed on a logarithmic scale. Table V shows that this similarity holds true for the accuracy rate, true positive and false positive rates of the two classifiers. This is due to both classifiers measuring the difference between two distributions for the same event.

VII. Discussion

The discussion is divided into two distinct sections. The first section will deal with the results from the fast-flux detection classifiers. The second section will focus on the results obtained for the classification of domain names.

A. Fast-flux detection

C5.0 decision trees have been used successfully in data mining and from the results it is observed that it is possible to generate decision trees capable of distinguishing between legitimate and fast-flux domains. The decision trees complement prior work by Holz *et al.* [8]and Perdisci *et al.* [7]. The same domain features where identified as those noted in their work as indicators of fast-flux domain behaviour. It is observed that decision trees are not capable of distinguishing between botnets that closely mimic CDNs and legitimate CDNs. Therefore it is proposed that a C5.0 decision tree classifier should be used in conjunction with other detection techniques. The C5.0 classifier does provide the benefit of being easily re-trained over time, as fast-flux domain behaviour shifts. This is opposed to manual heuristics, which need to be manually updated over time. Future extensions to the C5.0 classifier would include adding fuzzy logic to the decision process in an attempt to increase accuracy.

Using a Naive Bayesian classifier the limitations of standard decision trees were overcome. Leading to fast-flux networks that attempt to mimic the behaviour of legitimate CDNs being identified. The Naive Bayesian classifier achieved a higher true positive rate and lower false positive rate than standard heuristic based detection. Due to the statistical nature of the Bayesian classifier, accuracy increases with exposure to larger datasets. This behaviour ensures that the classifier is more adaptable to the changing techniques of hiding fast-flux domains. Eliminating the manual adjustment of the heuristics, as Holz *et al.* [8] stated would be required over time with their heuristic based detection system.

The proposed fast-flux detection techniques provide an additional layer of defence on the corporate network, allowing for the detection of botnet infested hosts on the network, which might have been missed by standard detection techniques. The active nature of the system furthermore allows for the identification of infected hosts, as opposed to domain blacklisting that simply blocks unwanted traffic but does nothing to mitigate future network access attempts by infected hosts.

B. Algorithmic name generation detection

The results indicate that it is possible to accurately classify domain names as either algorithmically generated or not. These statistical detection techniques allow for rapid classification of domain queries, which would normally require manual inspection of network traffic logs, long after the initial infection has already occurred. Furthermore, using statistical analysis allows for the detection of so called zero-day infections, which traditional detection signatures and blacklists have not been created for. The elimination of domain blacklisting is not proposed. Using statistical detection along with blacklists would provide better coverage and detection rates. This two factor system will increase the likelihood of mitigating the effects of network malware. By detecting algorithmically generated domain names, the system is capable of combating both current and future botnets that exhibit domain fluxing in an attempt to bypass network filters. A potential problem identified during testing was the use of algorithmically generated and non-standard domain labels by legitimate domains. Large domains, such as Google were identified to be using non-standard domain labels. An example

of this is *khmdb.google.com*. This can easily be overcome by including a whitelist into the detection system, where domains are labeled as safe if their second level domain is in the whitelist. In this example *google* would be added to the whitelist, ensuring that all future queries to *google.com* are marked as legitimate.

VIII. Conclusion

In this paper, techniques for detecting and mitigating botnet infections on a network were examined. These techniques aim to identify botnet behaviour without requiring system administrators to maintain blacklists or updated signatures. The use of statistical measures such as Naive Bayesian, Bayesian, Total Variation Distance and Probability for classifying domains as either malicious or legitimate were demonstrated. Analysis was performed on network traffic from a large South African University. It was determined that the use of a Naive Bayesian classifier provides a new accurate means of detecting both fast-flux domains and algorithmically generated domain names. Results show that Naive Bayesian classifiers provide the best level of accuracy with minimal false positives, followed by Bayesian classifiers and finally the Total Variation Distance classifier and Probability classifier providing similar results. The proposed solution provides an accurate means for improving network egress filtering. Furthermore providing an effective additional layer of network defence, usable in conjunction with existing defence systems. Future work will involve implementing supervised learning classifiers based on the current statistical measures.

References

[1] R. Bejtlich, *Extrusion Detection: security monitoring for internal intrusions*. Addison-Wesley, 2005.

[2] V. P. André Fucs, Augusto Paes de Barros, "New botnets trends and threats," in *Blackhat Europe*, 2007.

[3] G. Gu, J. Zhang, and W. Lee, "Botsniffer : Detecting botnet command and control channels in network traffic," *Technology*.

[4] S. Yadav, A. K. K. Reddy, A. N. Reddy, and S. Ranjan, "Detecting algorithmically generated malicious domain names," in *Proceedings of the 10th annual conference on Internet measurement - IMC '10*, New York, New York, USA, November 2010, p. 48.

[5] P. Porras, H. Saidi, and V. Yegnewaran, "Conficker c analysis," Tech. Rep., 2009.

[6] F. Leder and T. Werner. (2011) Containing conficker. [Online]. Available: http://net.cs.uni-bonn.de/uploads/media/c_domains_april2009.zip

[7] R. Perdisci, I. Corona, D. Dagon, and W. Lee, "Detecting malicious flux service networks through passive analysis of recursive DNS traces," *2009 Annual Computer Security Applications Conference*, pp. 311–320, December 2009.

[8] T. Holz, C. Gorecki, K. Rieck, and F. C. Freiling, "Measuring and detecting fast-flux service networks," in *MALWARE 2008. 3rd International Conference on Malicious and Unwanted Software, 2008*, 2008, pp. 24 – 31.

[9] Dragon Research Group. (2011) Team cymru. [Online]. Available: http://www.team-cymru.org/Services/ip-to-asn.html

[10] H. Xiong, P. Malhotra, D. Stefan, C. Wu, and D. Yao, "User-assisted host-based detection of outbound malware traffic," *University Computing*.

[11] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "Botminer: clustering analysis of network traffic for protocol- and structure-independent botnet detection," in *Proceedings of the 17th conference on Security symposium*, 2008, pp. 139–154.

[12] N. Jiang, J. Cao, Y. Jin, L. E. Li, and Z.-l. Zhang, "Identifying suspicious activities through DNS failure graph analysis," in *IEEE International Conference on Network Protocols (IEEE ICNP'10)*, 2010, pp. 144–153.

[13] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists : Learning to detect malicious web sites from suspicious urls," *World Wide Web Internet And Web Information Systems*, 2009.

[14] Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten, and I. Osipkov, "Spamming botnets: Signatures and characteristics," in *ACM SIGCOMM Computer Communication Review*, 2008.

[15] J. Zhang, X. Luo, R. Perdisci, G. Gu, W. Lee, and N. Feamster, "Boosting the scalability of botnet detection using adaptive traffic sampling," in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*. ACM, 2011, pp. 124–134.

[16] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A bayesian approach to filtering junk e-mail," in *AAAI'98 Workshop on Learning for Text Categorization*, 1998.

[17] R. Perdisci, G. Giacinto, and F. Roli, "Alarm clustering for intrusion detection systems in computer networks," *Engineering Applications of Artificial Intelligence*, vol. 19, no. 4, pp. 429–438, June 2006.

[18] Google. (2011) Top 1000 sites - doubleclick ad planner. [Online]. Available: http://www.google.com/adplanner/static/top1000/

[19] abuse.ch. (2011) Zeus monitor. [Online]. Available: https://zeustracker.abuse.ch/monitor.php?filter=level5

[20] ——. (2011) Spyeye monitor. [Online]. Available: https://spyeyetracker.abuse.ch/monitor.php?filter=level5

[21] ——. (2011) Fastflux tracker. [Online]. Available: http://dnsbl.abuse.ch/fastfluxtracker.php

[22] P. Royal. (2008) On the kraken and bobax botnets. [Online]. Available: http://www.damballa.com/downloads/r_pubs/Kraken_Response.pdf

[23] MalwareURL. (2011) Malwareurl. [Online]. Available: http://www.malwareurl.com/

[24] Rulequest Research. (2011) See5/c5.0. [Online]. Available: http://www.rulequest.com/see5-info.html

[25] P. S. David Hand, Heikki Mannila, *Principles of Data Mining*. The MIT Press, 2001.