Black Box Mimicry Attacks Against DNS Reputation Systems

A Thesis

by

Tillson T. Galloway

Approved by:

DocuSigned by:
Manos Antonakakis
96B58BF9FE46459...
Dr. Manos Antonakakis, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: 5/4/2022 | 1:37 PM EDT

DocuSigned by:
Angelos Keromytis
FFCE29BA237749E...
Dr. Angelos D. Keromytis
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: 5/4/2022 | 1:43 PM EDT

TABLE OF CONTENTS

Introduction

Alongside the recent explosion of internet communications, malicious activities have increased from bad actors, including black hat hackers, nation-state adversaries, and hacktivists. These attacks range from malware, spam, and covert attacks designed to infiltrate systems and to exfiltrate sensitive data. This has promoted the importance of fast and accurate attack detection in network security monitoring. Detecting an attack quickly and accurately can result in quicker remediation and reductions in monetary and data loss. In the last ten years, increases in computational power have led to the use of machine learning algorithms for this task. Algorithms have been designed using features extracted from data at all network layers.

Since 1985, the Domain Name System (DNS) has been an essential component of the internet. DNS is a system that provides a mapping between domain names that are easy to remember (e.g. google.com) and IP addresses that computers use to communicate (e.g. 30.3.5.2). These DNS mappings are called "records", and requests to DNS servers for records are called "queries." Malicious users commonly use DNS to aid their attacks: spammers typosquat popular domain names (e.g. go0gle.com or gooogle.com or google.tk) to enact phishing campaigns and malware generates hard-to-predict domain names that access command and control (C2) servers spread across the internet. This led to the creation of several systems designed to evaluate the reputation of DNS records.

DNS reputation systems are based on the analysis of DNS records and queries. DNS queries have been studied through active DNS data, where systems automatically request domains intending to collect longitudinal data, passive DNS data, where DNS requests from real users are monitored by a network sensor, and public zone data, where public data from DNS authorities is analyzed. A variety of machine learning techniques have been used for the task of classifying domains, including random forests, k-nearest neighbors, and recently, neural networks. Each of these techniques uses pattern recognition and statistics to detect malicious domains. With the techniques, however, come the advances of adversarial methods designed to poison and evade these models. While both generic attacks on machine learning models and specific attacks on common Machine Learning tasks like object detection have been achieved, there is a gap in attacks on network security algorithms. Recent research has shown that the narrower field of security provides ground for new techniques in adversarial attacks and creating models that provide robustness against them [9].

This project analyzes the robustness of two prominent models for DNS reputation, namely *Notos* [2], and *Exposure* [4], to practical adversarial attacks. In order to do this, we first re-created each model with two months of 2020 passive DNS (pDNS) data collected from a sensor at a public American university and a sensor at two large internet service providers (ISPs) and trained it on a set of malicious and benign data from public lists and filtered malware execution traffic. Using the models, we provide plausible techniques and algorithms to generate network traffic patterns that lead to model evasion and dataset poisoning. The intuition guiding the attack is that some features can be manipulated at a lower cost than others: for example, a feature involving whether a domain name with words in the English language can be manipulated at a lower cost than a feature involving usage of a shared-hosting provider.

This work is a meaningful contribution as it provides a general framework for adversarial attacks on network security models and the means by which they can be mitigated. Lever, et al. [7]

showed that analysis of malware infrastructure through Internet Protocol (IP) and DNS traffic can detect an infection several weeks before malicious binaries are captured and analyzed. As an example, attack attribution in the Mirai botnet attack was assisted by analysis of pDNS data [1]. The researchers utilized similar features as we use in this work (related historical domain names and IP addresses) to create DNS clusters that indicate operators of portions of the Mirai ecosystem. If the attackers manipulated the features presented by our paper, they may have been able to evade attribution using these methods.

Literature Review

Due to the explosion of network traffic in the last ten years paired with the increase of sophistication in adversaries, the field of network security and attack detection has become inseparable from Artificial Intelligence. IP, packet, and DNS data have all been analyzed at scale to detect adversarial behavior in datasets for purposes of attack attribution, detection, and prediction. Many of these experiments use DNS resolution data supplemented with BGP, whois, and known blacklists combined with machine learning algorithms (namely random forests, k-nearest neighbors, and neural networks) to detect malicious traffic [2, 3, 4]. While many techniques have been proposed for the detection of these malicious domains, there has been a lack of research in attempts to circumvent detection. Our goal is to fill this gap with a practical paper that offers methods that can be used both to attack DNS reputation models and to defend future models against attacks.

The *Notos* model[b] classified domains as malicious weeks before they began to appear in datasets using a combination of network-based (IP addresses, BGP), zone-based (TLDs, SLDs, 3LDs, and their lexicographical features), and evidence-based (appearances in known malicious datasets) features. Future models, including *Exposure* [4] expanded the model and increased its prediction accuracy by incorporating temporal features into the model, primarily monitoring the change in domain-to-IP relationships over time. These models achieved high precision and recall rates at their time of publication. In addition to detection using network-level sensors, there is also work in attack identification via analyzing large-scale public datasets, shown in *Kopis* [3]. These varying methods in models highlight a major difference in the cost of an attack on the respective model: an attacker would require a large number of resources, both computational and financial, in order to obtain a large enough set of network data to conduct a white box (full-knowledge) attack on Notos or Exposure. Additionally, deployment of the Notos or Exposure models is expensive as they require terabytes of recent network data to be effective. Kopis, on the other hand, is trained on public zone transfer data, making a model easier to generate and giving an attacker more knowledge.

Papernot et al. [8] divided attacks on machine learning algorithms into three categories: injecting into the data before the model is trained, attacking the model itself, and attempting to learn the model or the data used. Additionally, Papernot distinguishes between white box and black box attacks when attacking models. In a white box attack, the model, its parameters, and the training data are known by the attacker. In a black box attack, the model only exists as an oracle that can provide an answer—its learning parameters are not known to the attacker. Since then, more specific attacks have been proposed against Machine Learning algorithms, mainly neural networks [5]. These papers are paired with techniques to mitigate them, such as data augmentation or dropout. There seems to be a large gap in research specifically conducted on evading non-neural network models that are used for DNS-based network monitoring systems.

There has been recent work in creating models that are robust in avoiding these types of attacks. Neural networks are again the most popular, but there exists recent research on non-neural network algorithms and their robustness. *Notos*, *Exposure*, and *Kopis* were originally trained on random forests, so we specifically evaluate the literature on these. One technique, proposed by Yizcheng et al. [6], applies a monotonicity constraint on features where increasing the value of a feature strictly increases the probability that a sample is malicious, as given by the classifier. For example, for classifying phishing URLs, as the length of a URL increases, the classifier strictly increases its spam score. Monotonicity ideally forces an attacker to weaken their attack in order to evade a malicious classification. These techniques have been successfully applied to the problem of malware classification for PDF-based malware [6].

In recent years, many large-scale security incidents have used DNS, including the Mirai botnet and the 2020 SolarWinds attack. The importance of these DNS based models compared to analysis of malicious binaries is exemplified by Lever, et al. [7], who showed that analysis of malware infrastructure through IP and DNS traffic can detect infection on the scale of months before malicious binaries are dynamically analyzed. Therefore, the ability of an attacker to evade a DNS-based early detection system could lead to a longer period of infection on a computer before detection by intrusion detection or prevention systems.

Methods

In support of the primary goal of the work, to discover attacks that compromise the integrity of state-of-the-art DNS reputation models and to evaluate the models' robustness to attacks, we re-implemented the models with various datasets and feature extraction techniques. To create the models, we used a standard pipeline in machine learning: data collection, data cleanup, feature extraction, and model training.

Data Collection

The basis of each model was recursive DNS (rDNS) resolution queries. We ran our experiments on four datasets, each containing rDNS queries between August 1, 2020, and September 30, 2020. The three datasets came from different sources: passive rDNS from two large North American internet service providers (ISPs), and publicly available active rDNS data from the ActiveDNS project.

We obtained ground truth data from an aggregation of public and proprietary datasets. The benign dataset consisted of the top 10,000 entries in the Tranco popular domain list. The malicious dataset was populated by various public blocklists (PBLs). This included registered domains from DGArchive and Netlab360, PhishTank domains, the Spamhaus Domain Blocklist, and URLHaus domains. We also incorporated malware executions from a proprietary sandbox environment after cleaning the data by validating each execution with VirusTotal databases.

Data Cleanup

As our model was heavily based on the relationship between domain names, IP addresses, and TTLs over time, we considered only A and AAAA records, both of which resolve domain names to IP addresses. Additionally, because each of the models evaluated was released before the release of IPv6, we only consider IPv4 resolutions. We discarded resolutions containing unroutable, bogon IP addresses as they cannot host malicious data. Finally, we considered the effective second-level domain (e2LD, e.g. “google.com” in “maps.google.com”) for each domain and discarded those that were identified as a popular content delivery network (e.g. *.cloudfront.com), dynamic DNS (e.g. *.dyndns.org), or free shared hosting provider (e.g. *.github.io). Each of those domains is noisy and can contain both benign and malicious subdomains, which decreases the model’s accuracy.

Feature Selection

63 features from Notos [2] and Exposure [3] were computed from each dataset. The features were created based on the e2LDs of each of the resolutions in the original dataset.

Model Training

We evaluated the model on three popular Machine Learning models, namely random forests, k-nearest neighbors, and support vector machines. We performed a grid search over reasonable hyperparameters for each model and used ten-fold cross-validation for all training steps.

Due to the small size of the benign ground truth dataset, the resulting training data was unbalanced. To remediate this, we undersampled the larger dataset without replacement. We trained each model with 100 balanced samples and reported model statistics (accuracy, AUC, precision, recall) based on the average of the samples. The models trained on the two large passive DNS models both resulted in >99% precision and recall rates, a similar performance to *Notos* and *Exposure*.Table 1 shows the performance of the model for each dataset, which contains information about true positives, true negatives, false positives, and false negatives.

Table 1: Model performance for each dataset.

Dataset	**TPR**	**FPR**	**AUC**	**Accuracy**
Large pDNS	97%	4.4%	0.993	96.55%
Medium pDNS	98%	1.6%	0.997	94.26%
ActiveDNS	94.7%	5.9%	0.944	94.4%

.

Attacks

Our work differs from many attacks on Machine Learning models as the domain is highly constrained. In images, for example, an attacker attempting to fool an object classifier is able to manipulate each pixel over the full range of colors without any constraints. Additionally, to an attacker, each pixel is independent of one another. Features based on network structure, however, are highly correlated. Directing a domain at a single new IP address can have large effects on the feature values. For example, if the new IP address is a shared IP, the number of related domain names and IP addresses will increase significantly.

The attack presented was designed by manipulating the space of network traffic, which resulted in changes to the features vectors themselves. Additionally, this attack is based on a black box approach, meaning that an attacker does not have access to the model itself. The goal of our attack is to *evade* malicious classification by the model. We measure its success on its likelihood to evade the model, its financial cost, and its detectability by a network operator.

Mimicry Attacks

Mimicry attacks are based on the idea that a domain can hide malicious activity by mimicking benign traffic. This is achieved by routinely querying a popular domain (e.g. 'google.com'), storing its IP, and then changing the malicious domain's associated IP address to the benign IP. If this is performed many times, then the malicious domain will appear to have the same network structure as the benign one. The full algorithm is shown in Algorithm 1.

Algorithm 1 Black Box Mimicry Attack for malicious domain $\mathbf{d}_m$ using resource records of benign domain $\mathbf{d}_b$

Input: $\mathbf{d}_m, \mathbf{d}_b, n, t$

```
i ← 0
while i < n do
    (TTL, IP) ← Query A/AAAA resource record for d_b
    Set A/AAAA resource records for d_m to (TTL, IP)
    Sleep for t seconds
    i ← i + 1
end while
Set A/AAAA resource records for d_m pointing back to infrastructure controlled by the adversary.
```

Algorithm 1: The Mimicry Attack algorithm.

Results

We perform a mimicry attack on 250 spam and malware domains that each model correctly classified as malicious using 1000, 1500, and 2000 total resource records sampled from active DNS queries to the Tranco Top 5 (google.com, facebook.com, netflix.com, microsoft.com, facebook.com) domains. We first analyze the results for the 250 domains, and then give an in-depth analysis of a single domain that evaded the model. Table 2 shows the effectiveness of the attack.

Table 2: Evasion rate of black box mimicry attacks. Mimicry attacks were run on 250 correctly identified malware and spam domains from each dataset, each using the same 1000, 1500, and 2000 resource records sampled from active DNS records for google.com. The evasion rate of each set of samples is shown.

	1000 RRs inserted	1500 RRs inserted	2000 RRs inserted
Large pDNS Dataset	8%	35%	34.5%
Active DNS Dataset	88.3%	87.7%	85.7%

The mimicry attack is effective because domains and IPs controlled by large, legitimate entities have certain characteristics that a domain can emulate by adding those IPs to its own network traffic. This attack works particularly well in evading features that consider related historic domain names since manipulation of these features comes at a low financial cost. For a domain, *d*, a second domain, *q,* is considered to be a related historic domain name of *d* at least one of the IPs that *q* has resolved to has also been resolved to by *d*. Figure 2 shows an example of this relationship for three domains. In Figure 3, the first row shows a cumulative distribution function for the number of related historic domain names. The second row shows the average frequency of TLD in these related historic domain names.

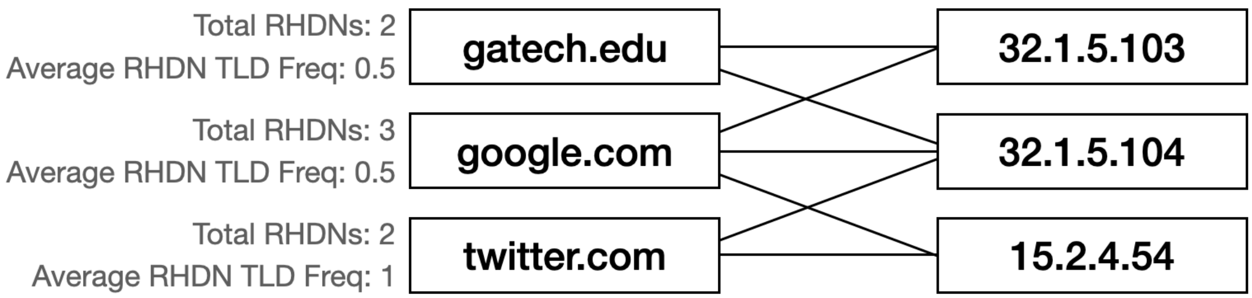
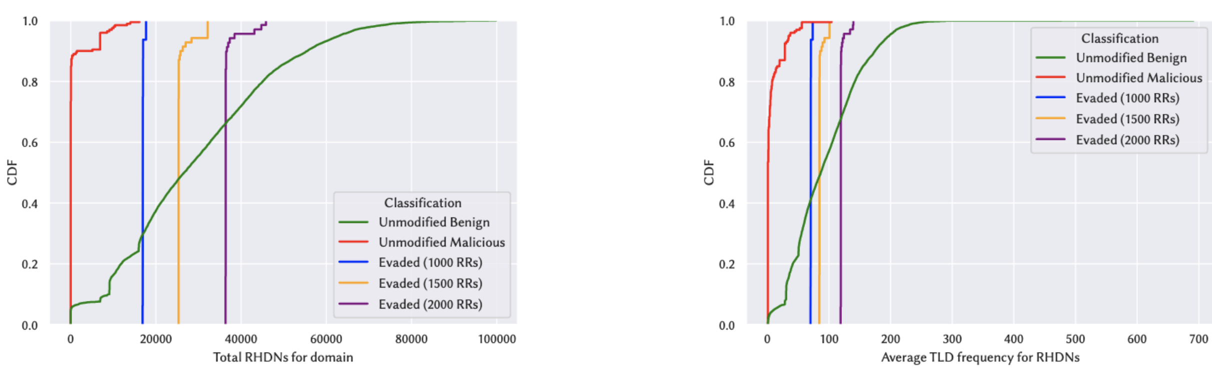


Figure 2: RHDN features for three domains. Both gatech.edu and google.com have resolved to the same IP address (32.1.5.103), so they are related historic domain names of each other. google.com and twitter.com are also related historic domains of each other, but twitter.com and gatech.edu are *not* related historic domains.



(a) Number of RHDNs for a domain **(b) Average frequency of each TLD in domain's RHDNs**

Figure 3: Comparison of cumulative distribution functions for two features after a black box Mimicry Attack where 250 domains mimic RRs from "google.com". (a) shows the number of related historic domain names. (b) shows the average frequency of all TLDs for each related historic domain name. In each case, adding more benign RRs results in a distribution more similar to the benign one**.**

Example

We demonstrate a specific example of this attack, starting from the malicious domain gkjexports<.>com. This domain successfully evaded the model after 1500 RRs from google.com were inserted.

To determine which features were most useful in the evasion, we use a technique developed by Xu et. al [10].The *impact score* of a feature is a measure of the change in a domain's reputation score by changing only that feature value. The impact score is determined by the change in reputation after overwriting values in the evaded sample to values of the original sample, and overwriting values in the original sample to values of the evaded sample. Table 3 shows the top ten impact scores for the example. Seven of the ten features are related to related historic domain names, indicating that those features are most vulnerable to this attack.

Table 3: Impact of each feature on evasion for gkjexports<.>com. "Original" and "Evaded" are the feature- values before and after a mimicry attack is performed. Δ_1 is the change in the original sample's reputation score after each of its features are overwritten with the evaded feature's value. Δ_2 is the change in the evaded sample's reputation score after each of its features are overwritten with the original feature's value. Impact is $\Delta_1 + \Delta_2$.

Feature	Original	Evaded	Δ_1	Δ_2	Impact
rhdn_count	2	36349	0.1398	0.19	0.3298
distinct_tld_count	1	305	0.0398	0.14	0.1798
stddev_tld_freq	0.0	1357.45	0.05	0.11	0.16
distinct_ips	1	2070	0.0407	0.08	0.1207
avg_3gram_freq	1.0	1.01	0.0	0.06	0.06
stddev_3gram_freq	0.0	0.09	0.0	0.03	0.03
access_ratio	4.91	10.04	0.01	0.01	0.02
daily_similarity	0.14	0.55	0.0	0.01	0.01
rhdn_length_stddev	8.49	8.91	0.01	-0.0	0.01
avg_tld_freq	2.0	119.18	0.01	-0.01	0.0

Discussion

The attacks performed significantly better on the active DNS dataset than the passive DNS ones. The most likely reason for this is a limitation in our feature engineering process. Active DNS queries many more e2LDs than longer FQDNs (e.g. "google.com" vs. "drive.google.com"), and each of our models only considers the e2LD when generating a feature. This causes domains with large numbers of subdomains to have significantly larger values for some fields (e.g. the number of RHDNs).

A limitation of this attack is that in practice, DNS reputation models are retrained on new data often (in the literature, this ranges from everyday to every two weeks). If the malicious IP or domain appears in one of the ground-truth datasets, this can result in the model overfitting to a previously successful evasion, therefore blocking it. It could also lead to a model learning a mimicry attack. This type of attack failure can be resolved by using ideas from domain generation algorithms (DGAs). DGAs are commonly generated using random strings (e.g. 8ajwefnc.com) or by concatenating English words (e.g. treedogcloud.com). The DGA is implemented both on the malware operator's infrastructure and in the malware itself: the operator's version registers new domain names and the malware's version updates its communications to use the new domain. When used alongside a mimicry attack, the malware operator's version registers the new domain and performs IP sampling against trustworthy domains for a fixed period. After this period ends and the model is poisoned with benign data relating to the domain, the domain begins resolving to malicious infrastructure, and simultaneously, the malware switches to this new domain. This continues iteratively, with new domains constantly being prepared by the DGA for future use.

Mimicry attacks are also limited by their high detectability: DNS analysts could easily discover an attack by creating an alert for new, non whitelisted domains that resolve to many different IPs in popular autonomous systems. Additionally, if the mimicked IPs are also being accessed in addition to being resolved, a network analyst may notice the additional traffic to their servers.

Conclusion

This thesis provides an overview of attacks on adversarial Machine Learning systems and provides an in-depth analysis of attacks on DNS Reputation Systems. We found that in a black box settings, we can construct examples that evade these models over 60% of the time.

Future Work

In future work, we could further explore the manipulation of DNS records to change the features by strategically taking certain actions over time. Xu et al. [10] proposed an evolutionary, black box algorithm that achieves 100% evasion on PDF-based malware classifiers by taking a pseudo random action, verifying that it preserves the malicious functionality, and then measuring the action's effect on the PDF's reputation score. After each batch of actions, the process is repeated with the actions that negatively affected the reputation score the most. This same process could be applied for DNS reputation systems by taking actions in the network space (registering an IP in a new autonomous system or country, changing the IP for a domain, etc.). Financial cost is easily measured as the sum of the cost of each action and detectability is measured by the number of actions that the attacker must take to achieve evasion.

By using our proposed attack on DNS reputation models, an attacker could successfully evade automatic detection by a machine learning based model. The work also provides a practical and empirical study on adversarial attacks with Machine Learning models with heavily constrained domains. We additionally provide grounds for further research into the robustness of these models, which will lead to an increase in attribution of attacks, quicker detection times, and a reduction of attacks altogether.

References

[1] Manos Antonakakis, Tim April, Michael Bailey, Matthew Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J. Alex Halderman, Luca Invernizzi, Michalis Kallitsis, Deepak Kumar, Chaz Lever, Zane Ma, Joshua Mason, Damian Menscher, Chad Seaman, Nick Sullivan, Kurt Thomas, and Yi Zhou. 2017. Understanding the Mirai Botnet. In *Proceedings of the 26th USENIX Conference on Security Symposium* (SEC'17), USENIX Association, Vancouver, BC, Canada, 1093–1110.

[2] Manos Antonakakis, Roberto Perdisci, David Dagon, Wenke Lee, and Nick Feamster. 2010. Building a Dynamic Reputation System for DNS. In *Proceedings of the 19th USENIX Conference on Security* (USENIX Security'10), USENIX Association, Washington, DC, 18.

[3] Manos Antonakakis, Roberto Perdisci, Wenke Lee, Nikolaos Vasiloglou, and David Dagon. 2011. Detecting Malware Domains at the Upper DNS Hierarchy. In *Proceedings of the 20th USENIX Conference on Security* (SEC'11), USENIX Association, San Francisco, CA, 27.

[4] Leyla Bilge, Sevil Sen, Davide Balzarotti, Engin Kirda, and Christopher Kruegel. 2014. Exposure: A Passive DNS Analysis Service to Detect and Report Malicious Domains. *ACM Trans. Inf. Syst. Secur.* 16, 4 (April 2014). DOI:https://doi.org/10.1145/2584679

[5] Yizheng Chen, Shiqi Wang, Yue Qin, Xiaojing Liao, Suman Sekhar Jana, and David A. Wagner. 2021. Learning Security Classifiers with Verified Global Robustness Properties. *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security* (2021).

[6] Yizheng Chen, Shiqi Wang, Dongdong She, and Suman Sekhar Jana. 2020. On Training Robust PDF Malware Classifiers. In *USENIX Security Symposium*.

[7] Chaz Lever, Platon Kotzias, Davide Balzarotti, Juan Caballero, and Manos Antonakakis. 2017. A Lustrum of Malware Network Communication: Evolution and Insights. In *2017 IEEE Symposium on Security and Privacy (SP)*, 788–804. DOI:https://doi.org/10.1109/SP.2017.59

[8] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. 2017. Practical Black-Box Attacks against Machine Learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security* (ASIA CCS '17), Association for Computing Machinery, Abu Dhabi, United Arab Emirates, 506–519. DOI:https://doi.org/10.1145/3052973.3053009

[9] Íñigo Íncer Romeo, Michael Theodorides, Sadia Afroz, and David Wagner. 2018. Adversarially Robust Malware Detection Using Monotonic Classification. In *Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics* (IWSPA '18), Association

for Computing Machinery, Tempe, AZ, USA, 54–63. DOI:https://doi.org/10.1145/3180445.3180449

[10] Weilin Xu, Yanjun Qi, and David Evans. 2016. Automatically Evading Classifiers: A Case Study on PDF Malware Classifiers. DOI:https://doi.org/10.14722/ndss.2016.23115