A Comprehensive Measurement-based Investigation of DNS Hijacking

Rebekah Houser
University of Delaware
rlhouser@udel.edu

Shuai Hao
Old Dominion University
shao@odu.edu

Zhou Li
University of California, Irvine
zhou.li@uci.edu

Daiping Liu
Palo Alto Networks
dpliu@paloaltonetworks.com

Chase Cotton
University of Delaware
ccotton@udel.edu

Haining Wang
Virginia Tech
hnw@vt.edu

***Abstract*—Attacks against the domain name system (DNS) have long plagued the Internet, requiring continual investigation and vigilance to prevent the abuse of this critical infrastructure. Among these attacks, DNS hijacking has repeatedly asserted itself as one of the most serious threats. In recent years, the severity of DNS hijacking has motivated renewed interest in developing more robust defenses. The size, dynamism, and diversity of the DNS ecosystem present nontrivial challenges to crafting an effective and scalable defense. Further, the relative rarity of documented DNS hijacking attacks makes them difficult to study in-depth. In this paper, we attempt to address the challenges in two thrusts. We first conduct an analysis based on the reports of confirmed DNS hijacking attacks and passive DNS records to characterize known DNS hijacking attacks and identify features for building defense mechanisms. Then we explore the extent to which the characteristic features can be used to build a DNS hijacking detection mechanism and evaluate its effectiveness from the perspective of a network gateway.**

***Index Terms*—DNS, Passive DNS, DNS Hijacking**

I. Introduction

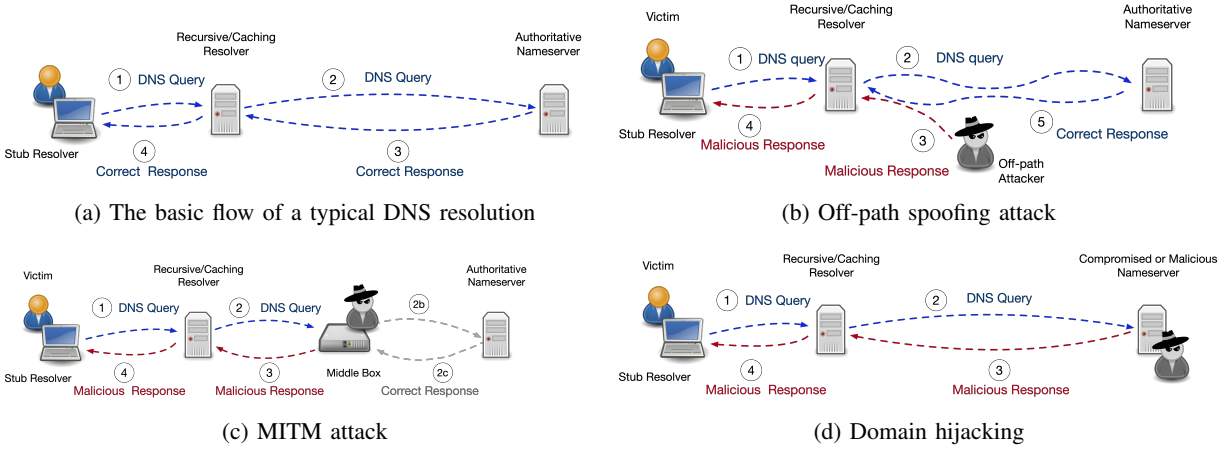
DNS hijacking attacks have garnered substantial attention over the past few years, leading to a renewed drive to create and strengthen defenses against this type of event. The Sea Turtle campaign has been somewhat of a catalyst in this area. It has prompted several groups involved in security research to publish details of the attacks, recommendations for defenses, and warnings that this incident could be a forerunner of new and increasingly serious DNS-focused attacks [4], [24], [30]. At the onset of the COVID 19 pandemic, some of these concerns resurfaced. With millions of individuals working from home, VPNs became even more vital to the day-to-day operations of many organizations. As the Sea Turtle campaign included VPNs as a primary target, this shift called attention to VPN security issues related to the DNS [33]. Given the interest in this area, we expect to see many new or improved methods of DNS hijacking detection and prevention researched and implemented over the next few years. Several challenges exist to work in this area, however. In this paper, we consider two of these.

The first challenge we address is a lack of clarity in threat models related to DNS hijacking. Efficiently addressing any attack requires a well-defined threat model, but in the case of DNS hijacking, we find some confusion surrounding factors necessary to construct such a model. The term "DNS hijacking" itself is a bit broad; we see it applied to an ISP's practice of redirecting NXDOMAIN responses [9], to manipulation caused by infected home routers reaching out to rogue DNS resolvers [53], or to attacks involving the unauthorized change of records in authoritative DNS servers [30]. Undertaking defenses against all of these with a single mechanism is unlikely to be practical. One must distinguish between the various flavors of DNS hijacking and understand how each can play a role in attacks leading to the compromise of protected resources.

The second challenge is the lack of in-depth research into certain types of DNS hijacking attacks. Although there are some, such as MITM attacks, that have been studied closely, others, such as domain hijacking are unpredictable, and short-lived, although they can have serious impacts even in that short time. These attacks are inherently difficult to study. Reports of them have appeared over several years, but, to the best of our knowledge, no work has examined them in-depth, or as a whole to evaluate what existing or potential defenses are most promising as mitigation.

In this work, we make the first attempts to address these challenges, in hopes of bridging the gap between the focus of academic research and the characteristics of real-world attacks. We present a taxonomy of DNS hijacking that aligns different attack vectors with the DNS infrastructure. Then, by skimming over security incidents from 2008 to 2020, we identified 34 relevant incidents from news stories, retrieved over 27,000 Indicators of Compromises (IOCs), and augmented them with passive DNS data from Farsight's DNSDB, which informs us when an attack happens, how records were changed by the attacker, *etc*. By conducting quantitative and qualitative analysis on this dataset, we identified features that help to detect the presence of DNS hijacking, whose effects were examined with the real-world attacks logged by the dataset.

The rest of this paper is organized as follows. Section II reviews the basics of DNS hijacking attacks. An analysis of the characteristics of known DNS hijacking attacks is presented in Section III. Section IV describes our experiments exploring



(a) The basic flow of a typical DNS resolution

(b) Off-path spoofing attack

(c) MITM attack

(d) Domain hijacking

Fig. 1: The flow of a typical DNS resolution and attack variants.

DNS hijacking detection. Limitations and future work are discussed in Section V, and related works are surveyed in Section VI. Finally, Section VII summarizes our work.

II. Background

The DNS is a distributed database containing the information users need to find the IP addresses of the hosts with which they wish to communicate. As shown in Figure 1a, to conduct a DNS resolution, a client known as a stub resolver first initiates a DNS query to a recursive resolver that will check its own cache for the answer to the query. If no cached answer is found, the recursive resolver will iteratively traverse the DNS hierarchy until receiving an answer from authoritative nameservers. That answer is then returned to the stub resolver.

The distributed nature of the DNS makes it scalable, but also creates challenges in ascertaining the integrity of responses. Because responsibility for domains is delegated to their owners, no party except the owner can know for sure when changes to a domain's DNS records are legitimate. Attackers exploit this situation and various weaknesses in the system to conduct DNS hijacking attacks. In such attacks, the attacker tricks end users into accepting incorrect responses to DNS queries, redirecting these users to servers of the attacker's choice.

A. DNS Hijacking Variations

DNS hijacking attacks tend to follow two basic approaches. The first involves infecting user devices and having them send DNS queries to malicious recursive resolvers. Attacks using this approach could be identified by checking DNS settings, or observing queries to unexpected or perpetually misbehaving resolvers [19], [49]. In the second approach, attackers convince a legitimate resolver – usually the recursive resolver – to accept malicious records. In this study, we focus on this second type of attack, which would be substantially harder to identify. Attacks that fall into this scenario follow a few different methods based on the position and capability of the attacker:

Off-path Spoofing. Off-path spoofing (illustrated in Figure 1b) refers to attacks in which an attacker cannot directly manipulate the traffic between a DNS resolver and nameservers but tricks the resolver into accepting a fake record. As long as the malicious record is cached, all parties using the resolver will be redirected when they send a query for the target domain. Most research into off-path spoofing attacks and defenses has focused on recursive resolvers. In the past two years, researchers have also demonstrated these attacks against stub resolvers and forwarders [6], [56].

Man-in-the-Middle (MITM). Between end users and nameservers, DNS communication may be manipulated by parties controlling the infrastructure traversed. Middlebox operations and redirection by both open and local recursive resolvers have been widely observed [10], [11], [32], [51]. Figure 1c illustrates this type of attack.

Domain Hijacking. Domain hijacking attacks have played out in a few ways. For example, an attacker may obtain unauthorized access to a registrar or a DNS management service and alter a domain's zone file. Alternatively, the attacker can leverage vulnerabilities in a registrar's processes to gain control over a domain's DNS records. Another approach may be to compromise a victim domain's account with a registrar or DNS management provider. Whatever the specific method, the result, as shown in Figure 1d, is that malicious answers appear to come from a legitimate authoritative nameserver.

B. Defenses

Means of preventing or detecting DNS hijacking attacks exist, but fail to adequately cover all relevant scenarios, particularly those involving domain hijacking. First, available defenses often are not or cannot practically be used. DNSSEC is the primary example [18], [22], but other measures, *e.g.*, strong passwords or two-factor authentication are often neglected, and some may be impractical (*e.g.*, registry locks [46]). Also, even after domain owners detect the issue and regain control of their domain, it may take days for malicious records to be expunged from resolver caches, during which time victims may continue

TABLE I: Hijacking Categories

Category	# Attacks	Description
Activism and Mischief	24	All of these are defacements, usually of popular websites. Of these 24, one third were defacements of regional versions of Google. One of these domains was defaced twice on separate occasions 5 years apart.
Malware and Spam Distribution	4	In 3 cases, domains were used to distribute exploit kits or other malware. In 1, domains were used to send spam.
Financial Gain	4	These attacks included 3 targeting domains related to cryptocurrency, and 1 targeting a bank.
Espionage Information Stealing	2	One case targeted a security firm, and the ultimate motivation may have been financial gain. One case (Sea Turtle) was apparently a state-sponsored hijacking.

to be exploited. This situation indicates the need for defense in depth. However, most defenses apply only to domain owners, and few are available to other stakeholders, such as network defenders.

In response to this scenario, we focus on assessing what methods parties other than the domain owner might use to defend against domain hijacking. We take the perspective of a defender whose goal is to protect resources within a local area network (LAN), *e.g.*, an enterprise network, and detect if changes in DNS responses entering the LAN indicate a DNS hijacking attack. In such a scenario, detecting DNS hijacking attacks against any and every domain is impractical, if not impossible. We consider that the defender will thus either monitor a fixed set of domains, or domains that become of interest due to the context in which they appear (*e.g.*, those from which mail is received). Thus, as we examine the problem of DNS hijacking attacks, we consider not only what features of these attacks may be used in detection systems, but what types of domains are targeted and how they are used.

III. Measurements and Analysis

In this section, we leverage the Indicators of Compromises (IOCs) gathered from known domain hijacking attacks to perform an in-depth analysis using passive DNS (PDNS) data. We investigate these attacks from three perspectives. First, we look for trends in the different groups of IOCs across attacks. Second, we take some general measurements to characterize patterns across multiple attacks. Finally, we examine a few individual attacks that are particularly noteworthy because of their impact or interesting characteristics they exhibit.

A. Dataset

We thoroughly examined the previous measurement studies that are related to DNS hijacking. Most of the research works we found were designed to measure the extent, causes, or impacts of censorship [3], [10], [11], [13], [14], [34], [36], [38]. Others measured more general DNS manipulation, usually focusing on NXDOMAIN redirection [31], [32], [51]. In news stories, we found primarily reports of domain hijacking attacks or attack campaigns. We identified 34 such incidents that occurred over a period of 12 years, from 2008 to 2020. We grouped these attacks into 4 categories according to the attackers' apparent motivation: activism or mischief, financial gain, distributing malware or spam, and stealing information or credentials. Table I briefly describes these groups. These motivations provide a framework for understanding some of the different behaviors observed in DNS hijacking attacks.

TABLE II: IOCs per Attack

	IP_A	NS_A	$FQDN_H$	$Apex_H$
Angler	454	0	22,571	5,249
Spammy Bear	1	0	4,007	4,007
Sea Turtle	33	5	30	21
Other	34	41	65	65

IP_A = Attacker IP addresses, NS_A = Attacker nameservers, $FQDN_V$ = Hijacked FQDNs, $Apex_H$ = Apex domain of hijacked FQDNs

We retrieved over 27,000 IOCs related to the domain hijacking incidents described above, summarized in Table II. Two campaigns, Angler and Spammy Bear, account for the vast majority of these. Many of the hijacked fully qualified domain names (FQDNs) were under the same apex domains. This was generally the case for Angler, where some domains had hundreds of subdomains in the lists. Thus, the number of hijacked domains (9,342) was much smaller than the FQDNs (26,673). To examine these attacks in detail, we retrieved data for the hijacked domains and nameservers from Farsight's Passive DNS database (DNSDB) [21].

B. Passive DNS

In this section, we provide definitions for the fields in the PDNS records that we retrieve. These are based on the definitions given in [20].

- **rrname**: the name of the domain for which information (*e.g.*, IP addresses) was requested
- **rdata**: the records returned in a response
- **rrtype**: the type of resource record included in a response (*e.g.*, `A`, NS, MX, or CNAME);
- **time_first**: the first time a response appeared in the PDNS dataset
- **time_last**: the last time a response appeared in the PDNS dataset
- **count**: the number of times a response has been captured by the PDNS sensors

Timestamps provide the epoch time at which a record was captured with seconds precision.

C. IOC Analysis

IOCs are analyzed according to the groups presented in Table II: victim FQDNs, attacker nameservers, and attacker IP addresses.

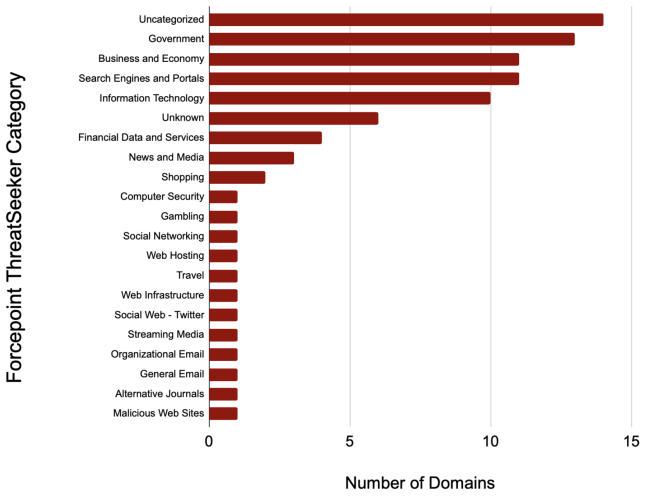


Fig. 2: Target Domain Category

Hijacked FQDNs. We first considered the characteristics of the hijacked FQDNs in order to understand what kinds of domains have been hijacked, and for what purpose. This information is relevant to threat modeling, as a defender may need to make decisions about what domains to monitor. We used VirusTotal [50] to collect ranking and category information for each apex domain. Most hijacked domains did not have a high rank. For Angler and Spammy Bear, the lists of hijacked domains included only a few (3 and 1, respectively) domains with an Alexa rank. Of these, the highest rank was 306,818. In addition, among the domains in other attacks, we found 53 that did appear in the Alexa list, although only 10 of these were in the top 1,000. All but one of these were targets of attacks involving defacement; in the other case, the domain was used to distribute malware [2]. This pattern highlights some expectations for the kinds of attacks against different types of domains. Attacks against popular domains are unlikely to remain undetected or unaddressed by their owners for very long. Therefore, the attackers need to be able to accomplish their goals in a short time. This scenario is fine for activists, as they can get the desired attention with relatively little effort, and in a short time. The scenario can also be effective for distributing malware, allowing attackers to have a high impact, albeit over a short time.

Figure 2 shows the Forcepoint ThreatSeeker categories for the hijacked domains, excluding those in the Angler and Spammy Bear attacks (of which the vast majority were "uncategorized" or "unknown"). Most domains (10) labeled as "uncategorized" are region-specific versions of popular domains under the ccTLDs `.cr` or `.nz`. Three are from the Sea Turtle campaign, and one was associated with a cryptocurrency wallet. All the domains in the government category were associated with Sea Turtle. The categories largely reflect two of the groups summarized in Table I: information gathering ("Government" category), and financial gain ("Business and Economy", "Financial Data and Services" categories), reinforcing the importance of these patterns as a consideration for building threat models.

```
;; first seen: 2013-08-27 20:20:13 -0000
;;  last seen: 2013-08-28 03:18:15 -0000
nytimes.com. IN NS ns1.syrianelectronicarmy.com.
nytimes.com. IN NS ns2.syrianelectronicarmy.com.

;; first seen: 2013-08-27 20:20:13 -0000
;;  last seen: 2013-08-28 03:18:15 -0000
nytimes.com. IN A 141.105.64.37

;; first seen: 2013-06-17 08:01:54 -0000
;;  last seen: 2013-08-28 02:11:40 -0000
ns1.syrianelectronicarmy.com. IN A 141.105.64.37

;; first seen: 2013-06-17 08:01:54 -0000
;;  last seen: 2013-08-28 02:11:41 -0000
ns2.syrianelectronicarmy.com. IN A 141.105.64.37
```

Fig. 3: Example of shared hosting between malicious DNS servers and web servers

Attacker Nameservers. We observe two general strategies for the nameservers used by attackers. In some cases, attackers replaced legitimate nameservers with their own hosts, while in others they leveraged DNS services provided by a third party. In cases where the attackers used their own nameservers, sometimes the new server's domain name clearly gave away the attack through an association with hacker groups (*e.g.*, `madleets.com`, `syrianelectricarmy.com`). In the case of Sea Turtle, the names were more subtle (*e.g.*, `cloudnamedns.com`, `lcjcomputing.com`). Both the PDNS data and accounts of the campaign suggest that the nameserver domains in this attack were either unregistered or unused (possibly parked) before the attack [4]. In both cases, a little research into the age or reputation of the nameservers should have shown them to be suspicious. This distinction may be much harder in the case where third-party nameservers are used. In one scenario, the attackers used nameservers from a provider that the hijacked domain was already using. Although two new nameservers did appear in the domain's NS records, they were under the same apex domain as those the hijacked domain was already using, so they do not appear particularly unusual. In this case, and at least three others, the attacker used Cloudflare nameservers. Because Cloudflare is a popular provider of various legitimate services, such attacks cannot be blocked simply based on a list or nameserver reputation, and may be very difficult to detect, even for a subject-matter expert. This scenario is not exclusive to Cloudflare, and we expect detecting such situations, if possible, would require an in-depth examination of information outside the DNS.

Attacker Infrastructure. We here consider whether attackers clearly favored certain networks as platforms for their attacks. We used RouteViews [41] BGP historic data to identify autonomous system numbers (ASNs) for networks used in the attacks and found 89 ASNs containing IPs used by rogue DNS servers or malicious hosts. Most ASNs (83) were associated with only one DNS hijacking attack, and none was associated with more than two. The results suggest no common trends between domain hijacking attacks in terms of the ASN used. External evidence of ASN reputation may be helpful, but relying on this indicator too much will likely lead to many false negatives.

Another interesting pattern that appeared in the attacker's use of IP addresses was related to shared hosting. That is, the attackers used the same machine to host both an authoritative DNS server, and a proxy or web server. (See Figure 3 for an example.) This sharing manifests itself in DNS records where both a domain and its authoritative nameserver resolve to the same IP address. The pattern was noted in reports of the Sea Turtle campaign [4], but we observed it in several other cases. Unfortunately, using this behavior as an indication of malicious activity is not straightforward. Investigating the nameservers for the hijacked domains we studied, we observed many legitimate cases of an apex or subdomain resolving to the same IP address as its associated nameservers within overlapping time frames.[1] There appeared to be two main scenarios for these cases: either the domain belongs to an organization that manages its own DNS servers, or the domain is relying on cloud providers. In both cases, the servers operated by the domain owner would be assigned IP addresses from the same block. Thus, this pattern of shared hosting is not inherently suspicious, and requires a more nuanced understanding of a domain's DNS deployment before it would be useful as a feature.

New rrnames. In the DNS hijacking scenarios we consider, the attacker cannot *directly* control what domains are queried. If the attacker is actively using the domains as part of a wider campaign to deliver malware (*e.g.*, Angler) or to distribute spam (*e.g.*, Spammy Bear), he or she can initiate queries indirectly. However, if the attacker is using the domain in a more passive manner, the attacker depends on users accessing the hijacked domain or its subdomains. Thus, where new *rrnames* appear in records associated with an attack, it may be helpful to understand how these are generated, and if this information can be leveraged to identify the attack.

To answer this question, for each hijacked domain, we checked for `A` records indicating an attack (*i.e.*, the item in the *rdata* field was in the list of attacker IOCs). We then checked if the FQDN in the *rrname* field had appeared in previous records, and if so, which records. Of the 4,258 `A` records manifesting attacker activities, the vast majority (4,149) were associated with `craigslist.org`. The domains `craigslist.com` and `craigslist.org` were hijacked in November 2014 and redirected to IP addresses under three ISPs not previously associated with the domains. As discussed below, this surge in the number of records appears to be closely tied to the attacker's use of a wildcard record. About 44% of FQDNs (1,864) appeared before the attack only in CNAME records. Of all the remaining instances, 431 contained an FQDN in the *rrname* field that had not been seen previously in the *rrname* field of any of the RRs. Subdomains of `craigslist.org` dominated all groups. In most cases, the CNAMEs previously used in resolving the given FQDNs were under domains distinct from these FQDNs. To receive requests associated with the given FQDNs, the attackers needed to

[1]We filtered out those cases where the nameserver was itself (*e.g.*, `ns1.example.com` for `example.com`).

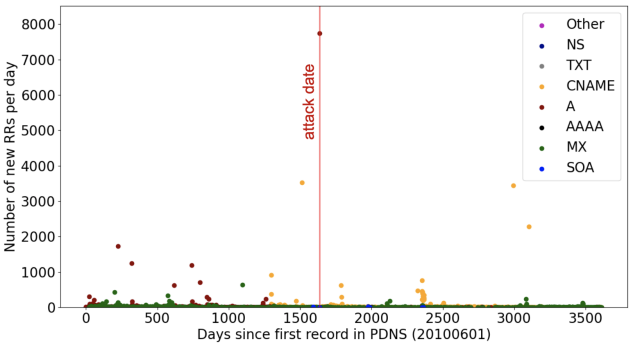


Fig. 4: Change in DNS patterns with attackers' use of wildcard

replace the CNAME records — thus the appearance of new `A` records. Disregarding instances associated with Craigslist, only two attacks appeared where the CNAMEs replaced were under the same domain as the associated FQDN. In one case, the day before the attack, new CNAME records appeared for some of these domains, redirecting them to the apex domain. It seems likely that the attack actually started a few hours earlier than indicated in our sources, and that these changes were initiated by the attacker. In the second case, there is no clear reason why the attacker chose to respond with `A` records rather than CNAMEs, since it controlled the resolution either way. In any case, this pattern of "centralizing" control by circumventing CNAME records is sufficiently common that we consider it worth evaluating when assessing changes to the DNS.

For those instances where the FQDN had not previously appeared in any records, we checked the reasons. Most of the cases were also related to the attack on Craigslist. The `A` records for the domains on the day of the Craigslist hijacking show that attackers created a wildcard record for the hijacked domain. This action suggests that while attackers wanted to ensure that they received all traffic, they were not sure what queries to expect, or they did not want to spend the time to build an extensive zone file. This led to a huge spike in the number of new `A` records seen for the first time that day (see Figure 4). This surge was mainly due to the appearance of queries for many domains that seem to be created by a DGA (Domain Generation Algorithm). While the source of these queries is unclear, it seems possible that such queries did exist before the date of the attack, and received NXDOMAIN responses (which do not appear in the PDNS dataset we used). Once the attack was initiated, queries for these domains received a legitimate response, thus inflating the number of new domains seen in the `A` records for the day.

Most of the other incidents were associated with the Sea Turtle campaign. Several of the subdomain names (`imap`, `pop`, `outlook`, *etc.*) highlight the fact that the attackers were interested in intercepting emails. The fact that these *rrnames* did not previously appear in the PDNS data does not mean they did not exist previously. It seems more likely that they existed, but were not visible. Since the Sea Turtle attacks generally involved the use of new authoritative nameservers, queries for these domains would have started following new

paths, possibly intercepting PDNS sensors for the first time. The same reasoning holds for two other attacks, targeting `google.ps` and `google.cr`, where the subdomains that appear redirected did not previously show up in records. In the one remaining case, the defacement of `nytimes.com`, the attackers created new nameservers under the hijacked domain: `sea4.nytimes.com` and `sea.nytimes.com`.

D. Case Studies

To examine these attacks in detail, we used a passive DNS dataset from Farsight. For three attacks no records were found, presumably because these attacks occurred before 2010, the first year in which records in the DNSDB were collected. The remaining 31 attacks included data for over 9,000 target domains, of which most (all but 68) were associated with Angler or Spammy Bear. We begin our analysis with a high-level view of certain attacks and their patterns.

Angler. Angler is an exploit kit that was active largely between 2011 and 2016, and made extensive use of domain shadowing. Domain shadowing involves creating subdomains under domains whose registrar accounts have been compromised, and using those subdomains for the attacker's purposes [15]. Based on the PDNS data for the domains targeted in the attack, it does not appear that the hijackings involved any change in nameservers. Indeed 4,155 of the 5,230 compromised domains in the list of Angler IOCs that yielded NS records used nameservers under only 1 apex domain. Spot-checking the PDNS data for 10 of the shadowed domains showed that in half the cases, no `A` records appeared prior to the attack, which is consistent with reports that these domains were largely dormant prior to the attack.

Spammy Bear. Spammy Bear is the name given to attackers responsible for campaigns running from mid-2018 to early 2019 that used hijacked domains to send "sextortion" messages and emails containing bomb threats with demands for ransom. According to reports of the incident, attackers did not actually compromise any accounts. Instead, they leveraged a weakness in a popular domain registrar to hijack about 4,000 domains [29]. Similar to the case of Angler, most (2,536 out of 4,001 with NS records) of the hijacked domains report nameservers under one apex domain, again indicating that the attackers did not change NS records. One interesting aspect of this attack was that attackers apparently created thousands of MX, SPF, DMARC, and DKIM records to support the spam campaign. This behavior highlights that certain defense measures based on the DNS become ineffective once an attacker controls the domain, suggesting the need for innovative detection methods.

Sea Turtle. The Sea Turtle campaign involved a series of domain hijacking attacks that appeared at least as early as 2017, and were still active in 2019. Attackers targeted domains for organizations that managed DNS or communication for other domains. These targets included government and private organizations largely in the Middle East and North Africa [5], [30]. Attacks were not constant, but periodically enabled [30]. The PDNS data for some of the domains reported to be hijacked show the hijackings occurring over a year apart, and the individual attacks lasting a relatively short period. Several of the hijacked domains appear to have used only the local or internal infrastructure to host their content prior to the attacks. The IPs used by the attackers belonged to cloud providers in other regions of the world. The attacks were evidently subtle enough to avoid immediate detection, since they continued to occur over an extended period. However, they stand out from the regular patterns of the PDNS data markedly because of the characteristics of the hijacked domains and the relatively low diversity in their IP addresses prior to the attacks.

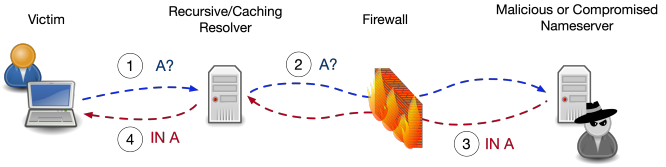


Fig. 5: Threat Model

Summary. While analysis of individual attacks provides helpful insights into the dynamics of those attacks, they also highlight that domain hijacking attacks are quite dissimilar in the patterns they create in DNS records. Some attacks may generate a spike in the number of responses or records for the target domain that appear in the data. Others might be so subtle as to avoid attention, even by domain owners (for a time at least). The behavior seen depends on factors such as the popularity of the domains, attackers' tactics, and how attackers leveraged the hijacked domains.

IV. DNS Hijacking Detection

A. Threat Model

The basis of our threat model is a defender whose goal is to detect if changes in DNS resolutions entering an enterprise network may indicate a DNS hijacking attack. As shown in Figure 5, the defender views DNS traffic between a local resolver and authoritative nameservers. This traffic would allow the defender to detect MITM or spoofing attacks directed against resolvers within the LAN. The defender specifically aims to detect DNS hijacking attacks directed against a set of domains of interest. These may include popular or high-risk domains, or those accessed frequently by users within the LAN. Since changes in domains' infrastructure are expected to occur periodically even under normal circumstances, detection is likely to require active measures extending beyond the DNS in some cases. Our goal is to examine if detection based on the DNS may be used to decisively flag at least some DNS hijacking attacks, or reduce the need for potentially expensive active probing (either automatic or manual).

B. System Design

In the system we explore, sensors capture DNS responses between a recursive resolver and authoritative nameservers. The system takes these responses as input, in addition to information from a block list, BGP Routing Information Base (RIB) data, and PDNS data. The PDNS data may include

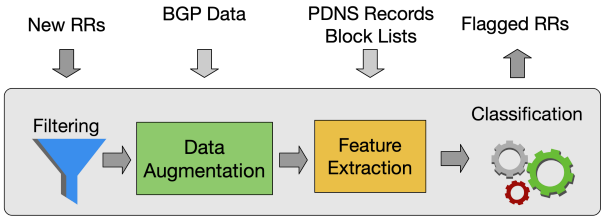


Fig. 6: DNS Hijacking Detection System

data collected within the network, data retrieved from third-party databases, or both. The system produces a subset of the observed responses that have been flagged as needing further verification. Figure 6 gives an illustration of the detector's stages, which we discuss in greater detail as follows.

Filtering. In the filtering stage, the system rejects responses that are incorrectly formatted or that can easily be marked as wrong answers. This filtering should implement standard checks against forged responses, such as those defined in RFC 5452 [26]. Also, the system should eliminate spoofed responses that can be identified with bailiwick checks [27]. Finally, responses with private IP addresses would be discarded.[2]

Data Augmentation. In the data augmentation stage, records are transformed to facilitate feature extraction. For NS, MX, and CNAME records, this includes adding a field indicating the apex domain of the item in the *rdata* field. For MX records, it will also add a field for the preference. The A records are the focus of this stage, however, as IP addresses are a rich source of information about the organizations and countries associated with a domain. The information added includes ASN, ASN owner and country, as well as the frequency with which an ASN has appeared in a public block list.

Feature Extraction. In this stage, augmented data is used to extract a set of features for the changes to be classified. An instance in our system comprises all the new records seen for a given domain on a given day, and the statistics of this group of new records are extracted and then used as features. Section IV-C discusses the details of the features selected.

Classification. The classification module ingests feature vectors generated by the previous stage and produces a list of flagged responses. Different algorithms may be used for this stage. We explore the results using two fundamental models, Random Forest and SVM. Given that domains may have widely disparate profiles, this stage may actually consist of multiple classifiers, each of which handles different groups of incoming responses, based on the characteristics of the domain in the *rrname* field.

C. Generalizing Characteristic Features: Changes per Day

Many of the DNS hijacking attacks manifested themselves in changes to multiple RR types. That is, RRs of multiple types (usually A and NS) were changed on the date of the hijacking. To evaluate these patterns, we divided the records by the date of their first seen timestamp. Excluding the domains targeted in the Angler and Spammy Bear Campaigns, for 55 domain names we identify 81 attack dates that appear in the dataset (attack dates are identified if the date is given in a report of the attack, or an attacker IOC appears in the in a domain's records on that date). On only 35.0% of these dates, no more than one new record type appeared. For all 68 hijacked domains with data (including those for which we found no attack dates), of the dates in which no attack appeared, 62% (out of over 55,300 measured) involved only one record type.

TABLE III: Attack Type Counts and IOCs

Attack Date		Not Attack Date	
RRTYPES changed	% Days	RRTYPES changed	% Days
A	29.63	A	32.23
NS A	20.99	A AAAA	19.69
NS CNAME A MX	11.11	CNAME	14.93
NS CNAME A	6.17	AAAA	7.09
NS A SOA	4.94	A CNAME	6.01
A AAAA	3.70	MX	5.69
NS	3.70	A MX	1.88
NS A AAAA SOA	3.70	A AAAA CNAME	1.56
NS A MX	2.47	AAAA MX	1.52
NS CNAME A AAAA SOA	2.47	NS SOA A AAAA CNAME	0.82
NS A MX SOA	2.47	CNAME MX	0.81
NS CNAME A AAAA MX SOA	2.47	NS A CNAME	0.70

Table III shows the top 12 RRTYPE groups for attack dates and normal dates on which no attack has occurred. Since most analyses of attacks focus on the A and NS records, we explored if and how attackers are using other types of records. We focused on MX, SOA, AAAA and CNAME RRs. A brief inspection of a few cases suggested that MX and CNAME records are used to point mail and popular subdomains (*e.g.*, www) to the apex domain, whose IP address has already been changed. The preference of new MX records was sometimes changed to 0 (highest priority), even when that preference may have never been used before in the domain's MX records. In the SOA record, the names in the RNAME and MNAME fields were often under the same domain as that of malicious nameservers.

In our detection, features are calculated in two steps. Given a domain, dom_i, and date day_j for which we would extract features, we first measure statistics of the DNS records that appeared for that domain on the day we wish to evaluate and for all days within a year prior to that date. These statistics include, for A, NS, and MX records, the number of records of each type that appeared that day, the number of new FQDNs or IP addresses in the *rdata* fields, and the number of domains, ASNs, ASN owners, and countries associated with those FQDNs and IP addresses. In addition, we also track the preference of the new MX records and monitor whether the new NS records are attached to the domain itself (or its subdomain). More generally, we keep track of how many record types appeared in the new data for the day, and how many *rrnames* were in the records of all types (excluding CNAME records). Having measured these statistics, we then

[2]Responses containing private IP addresses may indicate other attacks (*e.g.*, DNS rebinding). We ignore them, as they are not within our threat model.

TABLE IV: Features. Each of the features represents statistics of the *RRs* that appear for a domain on a particular date.

Feature Group	Description	Count
New A Record Features	New A *RRs* Previously unused IP addresses in the new A *RRs* Previously unused countries associated with the IP addresses in the new A *RRs* Previously unused ISPs associated with the IP addresses in the new A *RRs* The number of ASNs associated with the IP addresses in the new A *RRs* used by malware	5
New NS Record Features	New NS *RRs* Previously unused nameservers in the new NS *RRs* Previously unused domains for nameservers in the new NS *RRs* New NS *RRs* for the apex domain New NS *RRs* for a subdomain	5
New MX Record Features	New MX *RRs* Previously unused mail server in the new MX *RRs* Previously unused mail server domains in the new MX *RRs* Minimum mail server preference in the new MX *RRs*	4
Previous *RR* Featuress	Number of previously seen A *RRs* Number of previously seen NS *RRs* Number of ISPs in previously seen A *RRs* Number of nameserver domains in previously seen NS *RRs*	4
General Features	New *RRs* (not A, NS, MX or CNAME) and new *rrnames*	2

compare the statistics for the domain of interest against those of the 21 preceding days. If we find we don't have at least 21 samples in those 21 days, we expand the window to 42 days, then to all previous days. For each statistic measured, the associated feature is obtained by finding the difference between the value of that statistic on day_j with the median or minimum value of a statistic in previous days, and normalizing by the maximum of the two values.

Along with the characteristics identified in Section III-C, Table IV summarizes all the features we use to explore the DNS hijacking detection.

D. Classifiers

We evaluated two classification algorithms, Random Forest and SVM, with a 10-fold cross validation.[3] For the classifiers, we used Scikit-learn version 0.23.2 [39]. We used Scikit-learn because its simplicity, thorough documentation, and flexibility made it ideal at this point in our research, as we were interested in efficient exploration of simple models. In the future, tools that provide benefits such as greater scalability (*e.g.*, MLib [35]) or statistical analysis (*e.g.*, Statsmodels [45]) would likely provide more powerful options. Exploring these would be an interesting area of future work.

Random Forest classifiers are "ensemble methods" which combine the results from several simpler, less robust classifiers to obtain a final prediction. The algorithm in an SVM essentially attempts to find a boundary in a given feature space such that the boundary efficiently separates instances of different classes. In our work, we used the Scikit-learn SVC (C-Support Vector Classification) classifier specifically.

The Scikit-learn implementations of Random Forests and SVMs have several parameters. In our case, all of these except class weights were left at their default values. Here we review a few of the most relevant parameters here. A more thorough discussion is available in the Scikit-learn documentation [39]. For Random Forests:

- **n_estimators** is the number of decision trees whose predictions are combined to obtain the final result. The default value is 100.
- **criterion** is the approach used to quantify how well a split divides the data. The default criteria is gini impurity.
- **bootstrap**, if set to true, indicates that instances are to be sampled to build decision trees. Otherwise, all instances are used for each tree. The default is true.
- **max_features** sets the number of features to consider when determining how to split data. The default is to use the square root of the total number of features.

Some of the most relevant parameters for the SVM we used are as follows:

- **kernel**: The kernel function used. The default is the Radial Basis Function (RBF).
- **C**: Regularization is inversely proportional to C. The default value is 1.
- **gamma**: The kernel coefficient. The default value is 1/(*number of features*$*$*variance of the input data*).

E. Experiment

We here examine how the features we examined can be leveraged to build a detection model for DNS hijacking at a LAN gateway.

1) Dataset: Our dataset consists of PDNS records from the Farsight DNSDB. In addition to the data for domains known to have been hijacked, we also retrieved PDNS records for 816 other domains which were selected to represent three groups: Alexa Top, Alexa Business, and Alexa Regional domains. The Alexa Top domains include 96 of the Alexa top 100 [7]. Four of the domains in the list of hijacked domains were also in the top 100 from the list; we generally treat these separately from the rest of the Alexa top 100. Alexa Business domains were taken from the categorized Alexa Top Sites lists [8]. The Alexa Regional domains were domains with ccTLDs that appeared

[3]In early tests, we also explored using unsupervised learning but found this yielded poor results, so did not fully develop this approach.

in the Alexa top list and that had a corresponding domain with a generic TLD in the top 100 (*e.g.*, `google.com.bd`). We retrieved over 650 million unique records (*rrname, rrtype, rdata*) tuples covering a period of 10 years (2010-2020).

2) Approach: With the process of filtering and augmenting the data (Section IV-B), we label the dataset and extract the features to build the detector. In our experiments, one instance represented the changes in DNS records that occurred for a single domain on a single day. Thus, we evaluated changes on a per-day, per-domain basis. Labeling the occurrence of domain hijacking was straightforward using reported attack dates and checking where known IOCs appeared in records. In order to identify benign instances, we used the following approach for each domain, *D*, in our dataset:

- Find "trusted" nameservers, mail servers, and CNAMEs. For example, in an NS record for *D*, we first extracted the apex domain of the nameserver. If that domain appeared in NS RRs spanning a period of over 26 weeks, we marked all nameservers under that apex as *trusted* for *D*. We used the same process for MX and CNAME RRs.
- Find "trusted" ASN owners. For each `A` record, using MaxMind [1], we identified the ASn owner associated with the IP address in that record. If IPs belonging to the ASN owner were seen in `A` records spanning a period of over 26 weeks, we marked all records with IPs associated with that ASN owner as trusted for *D*.
- For each instance, if *all* NS, CNAME, and MX domains and ASN owners in new records are trusted for *D*, we marked the day as *benign*.

Any instances that could not be labeled as attack or benign dates were marked as *unknown* and not used for validation.

3) Results: For evaluation, we divided the instances related to attacks into three groups by the year in which they occurred: 2010-2012 (15 instances), 2013-2015 (17 instances), and 2016-2021 (47 instances). For each of these time periods, we also sampled 10,000 benign instances from the same time period. We ran the test with each group using stratified cross-validation, so that malicious instances were divided between training and test folds in each round of validation.

For attacks spanning multiple days, features derived for the second and subsequent days incorporate information from the first day, contributing to false negatives. To explore the impact of these cases, we assessed the results if we assumed such attacks were detected the first day, thus allowing the prevention of further incidents. If our classifier detected a hijacking for a domain on a given day, we checked for attacks involving that domain in the following 21 days. If any such instances exist and were not detected, we calculate performance (precision, recall, and false negatives) as if those cases had been identified.

Table V provides details of classifier performance. The false positive rate (FPR) is between 0.02% and 0.08%, and the false negative rate (FNR) is between 12% and 30%. These results are more than adequate, especially given that the system we have designed is intended to be a first step prior to additional probing. We note that since most of the false positives involve legitimate changes in the DNS, it might be possible to verify changes easily, simply by checking if previously used nameservers are still operating, and if so, do they agree with new nameservers.

TABLE V: Results of cross validation

	Years in Dataset	Precision*	Recall*	AUC -PR	FPR	FN*
Random Forest	2010-2013	0.85 (0.86)	0.73 (0.8)	0.85	0.02%	4 (3)
	2013-2016	0.85 (0.88)	0.65 (0.82)	0.73	0.02%	6 (3)
	2016-2020	0.87 (0.89)	0.57 (0.7)	0.71	0.04%	20 (14)
SVM	2010-2013	0.88 (0.88)	0.93 (1.0)	0.67	0.02%	1 (0)
	2013-2016	0.72 (0.76)	0.76 (0.94)	0.73	0.05%	4 (1)
	2016-2020	0.7 (0.75)	0.4 (0.51)	0.5	0.08%	28 (23)

*Values in parentheses show results when assuming early detection of multi-day attacks, as described in Section IV-E3.

Most of the false negatives we identified were associated with the Sea Turtle campaign. Further inspection reveals a good bit of "noise" within the records, possibly associated with other attacks. In some cases, this noise includes large numbers of NS records where the *rrname* or *rdata* field look like something created by DGAs. In others, the noise comes from known or suspected hijacking attacks. Both cases could contribute to false negatives. To be conservative, we have purposefully *not* attempted to remove other attacks from the data when building features, and the measurements from these days will affect those of subsequent days. Such an approach likely gives a worse case scenario, and these results could likely be improved by using more rules to filter previous attacks when extracting features that involve comparisons between new and old records.

4) Feature Importance: We investigate feature importance using permutation importance [39]. We observed that both the SVM and Random Forest classifiers rely heavily on the feature indicating if new NS records have appeared for the apex domain. The SVM relies almost entirely on this feature, while the Random Forest includes several others, including the change in the number of new ASN owners, and how many nameservers the domain had used previously. As such, we conclude that the only consistent discernible differences between most changes in the DNS and those caused by DNS hijacking attacks involve changes in nameserver. While we anticipated NS changes would be important, it appears they are definitive, and that improving automatic detection will involve further characterizing NS changes.

V. Discussion

A. Limitations

In our study, both our experiments and proposed defense strategies depend heavily on the appearance of new nameservers and AS owners in a domain's DNS records. However, an attacker may be able to gain access to nameservers under the same apex domain as those used previously by a domain. For example, the attacker could leverage third-party platforms to hide the true identity or location of its own nameservers.

Further, given the growing adoption of cloud platforms, an attacker may more easily gain access to machines located in the same AS or belonging to the same ISP as that of the hijacked domain, resulting in the ineffectiveness of the proposed features. We note, however, that even a human expert would likely be unable to identify such attacks based simply on DNS traffic, so we do not see this as a significant shortcoming of our approach.

The performance of our system in cross-validation was satisfactory, but it may require further development to be usable in a real-world setting. In particular, we have little information on the scalability or latency of the system. Our main goal of this detection experiment is to evaluate if the features we derived based on previous hijacking attacks are useful, and understand what additional features might be needed. Building on these insights, we will compare our approach with those of other systems in future work.

B. Ethical Considerations

In our experiments, we considered ethical issues surrounding the use of PDNS data. Concerns about passive DNS mainly focus on users' privacy, although issues of internal DNS disclosure or zone reconstruction are also important [16], [48]. As noted in [48], a well-configured PDNS collection will not infringe on individuals' privacy. In any case, the data that we receive from Farsight has been stripped of all client and server IP addresses, making it impossible for us to associate DNS traffic with users or locations. Regarding privacy issues related to the domains themselves, we do not attempt to leverage the data to obtain information not immediately relevant to the threat we study. Where we have analyzed a domain's DNS usage in depth, our purpose has been to attempt to identify attacks and patterns that can be generalized to describe normal changes, and we do not construct detailed models of a domain's infrastructure.

VI. Related Work

While many works have explored a single type of DNS hijacking attack or a specific tactic, few have considered these collectively and clarified the distinctions. In the wake of Kaminsky's disclosures regarding the potential of cache poisoning attacks, several works focused on addressing off-path spoofing [17], [23], [28], [37], [54]. Many have measured common MITM attacks [3], [10], [11], [13], [14], [34], [36], [38], usually in the context of studying censorship or detecting NXDOMAIN redirection.

A few works have highlighted the distinctions between different types of domain hijacking. In [23], authors note that identifying whether an attacker is using off-path spoofing or an MITM approach is the first step in evaluating defenses against cache poisoning. They do not explicitly consider domain hijacking. In [47], authors mention all three types of attack that we have examined, but do not address distinct characteristics of these. Finally, in [52], the author addresses different root causes for incorrect responses from authoritative nameservers. This primarily focuses on errors rather than attacks.

Several works have proposed novel approaches or features for detecting bogus DNS responses. In [49], researchers identify misbehaving resolvers used by clients within a network. Their features largely focus on frequency distributions of DNS response parameters. Systems relying on historical DNS and agreement between multiple resolvers to detect DNS manipulation are proposed in [40], [54], [55]. These systems could all at least theoretically detect off-path spoofing and MITM attacks, but are not designed to address domain hijacking.

In contrast, the system in [28] leverages historical DNS logs from a single vantage point, and Whois records to identify potential domain hijacking attacks. Their heuristics-based approach relies primarily on checking if unusual responses did arrive from an appropriate authoritative nameserver, and that the nameserver itself can be validated via Whois records. Although some domain owners chose to change their authoritative nameservers without updating their Whois records [42], this approach would likely be effective. However, the use of Whois records suggests this approach is somewhat unscalable, and is perhaps best applied to cases that have already passed other filters for suspicious behavior.

Another system that could, by design, detect various types of DNS manipulation is Anax [12], which relies on background information extracted from PDNS datasets to evaluate new changes. The features Anax uses largely focus on the diversity of domains resolving to IPs in a BGP prefix, and on what portion of these are associated with CDNs. Authors in [44], follow a similar approach to evaluating whether an IP address should be trusted for a particular domain. Their approach is based on similarity between that domain and others resolving to the IP address in question. The features and filters we have proposed are complementary to their approach, as we consider mechanisms and features beyond those that can be extracted from A records, and information regarding IP addresses.

VII. Conclusions

In this work we extensively studied the characteristics of DNS hijacking attacks and explored the detection of such attacks from the position of a party defending a local network from attacks originating outside the network, including off-path spoofing, MITM, and domain hijacking attacks. We analyze previous studies or reports of known attacks. Based on measurements related to these, we derived a set of features that might be used to identify unusual changes in a domain's DNS that require further inspection or blocking. We tested our approach on a large passive DNS dataset containing several million records collected for a period of over 10 years. The results of validation and testing have a low FPR, consistently less than 1%. Examining feature importance highlights the importance of focusing on nameserver changes, suggesting a promising area for future work.

Acknowledgements

This material is based upon work supported in part by the NSF Graduate Research Fellowship Program under Grant No.

1247394, NSF DGE-1821744 and CNS-2047476, and a gift from Cisco. The data used was provided by a research grant from Farsight.

REFERENCES

[1] "GeoIP2 Databases." [Online]. Available: https://www.maxmind.com/en/geoip2-databases

[2] L. Abrams, "Popular Anime Site Crunchyroll.com Hijacked to Distribute Malware," Nov. 2017.

[3] G. Aceto, A. Botta, A. Pescapè, N. Feamster, M. Faheem Awan, T. Ahmad, and S. Qaisar, "Monitoring Internet Censorship with UBICA," in Traffic Monitoring and Analysis (TMA), 2015.

[4] D. Adamitis and P. Rascagneres, "Sea turtle keeps on swimming, finds new victims, DNS hijacking techniques," https://blog.talosintelligence.com/2019/07/sea-turtle-keeps-on-swimming.html, 2019.

[5] D. Adamitis, D. Maynor, W. Mercer, M. Olney, and P. Rascagneres, "DNS Hijacking Abuses Trust In Core Internet Service," https://blog.talosintelligence.com/2019/04/seaturtle.html, 2019.

[6] F. Alharbi, J. Chang, Y. Zhou, F. Qian, Z. Qian, and N. Abu-Ghazaleh, "Collaborative Client-Side DNS Cache Poisoning Attack," in IEEE INFOCOM, 2019.

[7] http://s3.amazonaws.com/alexa-static/top-1m.csv.zip, Amazon.

[8] https://www.alexa.com/topsites/category, Amazon.

[9] N. Anderson, "Comcast adopts DNS hijacking, imposes irritating opt-out," https://arstechnica.com/tech-policy/2009/08/comcasts-dns-redirect-service-goes-nationwide/, Aug. 2009.

[10] Anonymous, "The Collateral Damage of Internet Censorship by DNS Injection," ACM SIGCOMM CCR, vol. 42, no. 3, 2012.

[11] ——, "Towards a Comprehensive Picture of the Great Firewall's DNS Censorship," in USENIX FOCI, 2014.

[12] M. Antonakakis, D. Dagon, X. Luo, R. Perdisci, W. Lee, and J. Bellmor, "A Centralized Monitoring Infrastructure for Improving DNS Security," in Recent Advances in Intrusion Detection (RAID), 2010.

[13] S. Aryan, H. Aryan, and J. A. Halderman, "Internet Censorship in Iran: A First Look," in USENIX FOCI, 2013.

[14] E. Athanasopoulos, S. Ioannidis, and A. Sfakianakis, "CensMon: A Web Censorship Monitor," in USENIX FOCI, 2011.

[15] N. Biasini and J. Esler, "Threat spotlight: Angler lurking in the domain shadows," Mar. 2015.

[16] S. Bortzmeyer, "DNS Privacy Considerations," IETF RFC 3568, 2015.

[17] S. Y. Chau, O. Chowdhury, V. Gonsalves, H. Ge, W. Yang, S. Fahmy, and N. Li, "Adaptive Deterrence of DNS Cache Poisoning," in Security and Privacy in Communication Networks, 2018.

[18] T. Chung, R. van Rijswijk-Deij, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson, "Understanding the Role of Registrars in DNSSEC Deployment," in ACM Internet Measurement Conference (IMC), 2017.

[19] D. Dagon, C. Lee, W. Lee, and N. Provos, "Corrupted DNS Resolution Paths: The Rise of a Malicious Resolution Authority," in NDSS, 2008.

[20] A. Dulaunoy, A. Kaplan, P. Vixie, and H. Stern, "Passive DNS - Common Output Format," Working Draft, IETF, Tech. Rep., June 2017. [Online]. Available: http://www.ietf.org/internet-drafts/draft-dulaunoy-dnsop-passive-dns-cof-03.txt

[21] "Passive DNS historical internet database: Farsight DNSDB," https://www.farsightsecurity.com/solutions/dnsdb/., Farsight.

[22] S. Hao, Y. Zhang, H. Wang, and A. Stavrou, "End-Users Get Maneuvered: Empirical Analysis of Redirection Hijacking in Content Delivery Networks," in the 27th USENIX Security Symposium, 2018.

[23] A. Herzberg and H. Shulman, "Antidotes for DNS Poisoning by Off-Path Adversaries," in AERS, 2012.

[24] M. Hirai, S. Jones, and B. Read, "Global DNS Hijacking Campaign: DNS Record Manipulation at Scale," Jan. 2019. [Online]. Available: https://www.fireeye.com/blog/threat-research/2019/01/global-dns-hijacking-campaign-dns-record-manipulation-at-scale.html

[25] R. Houser, "Investigations of the Security and Privacy of the Domain Name System," Ph.D. dissertation, University of Delaware, 2021.

[26] A. Hubert and R. van Mook, "DNS Resilience against Forged Answers," RFC Editor, Tech. Rep. RFC5452, Jan. 2009.

[27] "ISC Passive DNS Architecture," https://www.farsightsecurity.com/assets/media/download/passive-dns-architecture.pdf, Internet Systems Consortium, Inc., 2012.

[28] A. Kalafut and M. Gupta, "Pollution Resilience for DNS Resolvers," in IEEE ICC, 2009.

[29] B. Krebs, "Bomb Threat, Sextortion Spammers Abused Weakness at GoDaddy.com," https://krebsonsecurity.com/2019/01/bomb-threat-sextortion-spammers-abused-weakness-at-godaddy-com/, Jan. 2019.

[30] ——, "A Deep Dive on the Recent Widespread DNS Hijacking Attacks," Feb. 2019. [Online]. Available: https://krebsonsecurity.com/2019/02/a-deep-dive-on-the-recent-widespread-dns-hijacking-attacks/

[31] C. Kreibich, N. Weaver, B. Nechaev, and V. Paxson, "Netalyzr: Illuminating the Edge Network," in ACM Internet Measurement Conference (IMC), 2010.

[32] M. Kührer, T. Hupperich, J. Bushart, C. Rossow, and T. Holz, "Going Wild: Large-Scale Classification of Open DNS Resolvers," in ACM Internet Measurement Conference (IMC), 2015.

[33] A. Kwan, "Five Security Blind Spots from Prolonged Implementation of a Business Continuity Plan Amid COVID-19," 2020. [Online]. Available: http://www.circleid.com/posts/20200225_five_security_blind_spots_from_prolonged_implementation_of_bcp/.

[34] G. Lowe, P. Winters, and M. L. Marcus, "The Great DNS Wall of China," New York University, Tech. Rep., 2007.

[35] MLib. [Online]. Available: https://spark.apache.org/mllib/

[36] Z. Nabi, "The Anatomy of Web Censorship in Pakistan," in USENIX FOCI, 2013.

[37] K. Park, V. S. Pai, L. Peterson, and Z. Wang, "CoDNS: Improving DNS Performance and Reliability via Cooperative Lookups," in OSDI, 2004.

[38] P. Pearce, B. Jones, F. Li, R. Ensafi, N. Feamster, N. Weaver, and V. Paxson, "Global Measurement of DNS Manipulation," in USENIX Security Symposium, 2017.

[39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[40] L. Poole and V. S. Pai, "ConfiDNS: Leveraging Scale and History to Improve DNS Security," in USENIX Workshop on Real, Large Distributed Systems, 2006.

[41] http://www.routeviews.org/routeviews/, RouteViews.

[42] J. S. Sauver, "What is a Bailiwick?" https://www.farsightsecurity.com/txt-record/2017/03/21/stsauver-what-is-a-bailiwick/, 2017.

[43] J. S. Sauver and P. Foremski, "A Decade of Passive DNS: a Snapshot of Top-Level Domain Traffic," 2021, https://info.farsightsecurity.com/a-decade-of-passive-dns, last accessed on 05/10/2021.

[44] W. Scott, T. Anderson, T. Kohno, and A. Krishnamurthy, "Satellite: Joint Analysis of CDNs and Network-Level Interference," in USENIX Annual Technical Conference (ATC), 2016.

[45] S. Seabold and J. Perktold, "statsmodels: Econometric and Statistical Modeling with Python," in 9th Python in Science Conference, 2010.

[46] SIDN, "Registry locks: great potential but little current demand," 2019. [Online]. Available: https://www.sidn.nl/en/news-and-blogs/registry-locks-great-potential-but-little-current-demand.

[47] S. Son and V. Shmatikov, "The Hitchhiker's Guide to DNS Cache Poisoning," in Security and Privacy in Communication Networks, 2010.

[48] J. M. Spring and C. L. Huth, "The Impact of Passive DNS Collection on End-user Privacy," in Securing and Trusting Internet Names, 2012.

[49] M. Trevisan, I. Drago, M. Mellia, and M. M. Munafò, "Automatic detection of DNS manipulations," in IEEE International Conference on Big Data (Big Data), 2017.

[50] VirusTotal. [Online]. Available: https://www.virustotal.com

[51] N. Weaver, C. Kreibich, and V. Paxson, "Redirecting DNS for Ads and Profit," in USENIX FOCI, 2011.

[52] D. Wessels, "DNS Cache Poisoners Lazy, Stupid, or Evil?" in NANOG. NANOG, 2006.

[53] G. Ye, "70+ different types of home routers (all together 100,000+) are being hijacked by GhostDNS," https://blog.netlab.360.com/70-different-types-of-home-routers-all-together-100000-are-being-hijacked-by-ghostdns-en/, 2018.

[54] L. Yuan, K. Kant, P. Mohapatra, and C. Chuah, "DoX: A Peer-to-Peer Antidote for DNS Cache Poisoning Attacks," in IEEE ICC, 2006.

[55] L. Yuan, C.-C. Chen, P. Mohapatra, C.-N. Chuah, and K. Kant, "A Proxy View of Quality of Domain Name Service, Poisoning Attacks and Survival Strategies," ACM Trans. Internet Technol., vol. 12, no. 3, May 2013.

[56] X. Zheng, C. Lu, J. Peng, Q. Yang, D. Zhou, B. Liu, K. Man, S. Hao, H. Duan, and Z. Qian, "Poison Over Troubled Forwarders: A Cache Poisoning Attack Targeting DNS Forwarding Devices," in USENIX Security Symposium, 2020.

TABLE VI: Hijacked domains

Group	Domains
Activism and Mischief	comcast.net (5/28/2008), icann.com, iana.com (6/26/2008), hsbc.co.nz, linux.co.nz, sony.co.nz, coca-cola.co.nz, xerox.co.nz, fanta.co.nz, f-secure.co.nz, windowslive.co.nz, bitdefender.co.nz, msn.co.nz, microsoft.co.nz, hotmail.co.nz, live.co.nz, msn.org.nz, msdn.co.nz (04/21/2009), twitter.com (12/18/2009), baidu.com (1/12/2010), secunia.com (11/25/2010), google.com.bd (1/8/2011, 12/21/2016), theregister.co.uk, telegraph.co.uk, ups.com, nationalgeographic.com, acer.com, betfair.com, ning.com (9/4/2011), google.ro, yahoo.ro, microsoft.ro, paypal.ro, kaspersky.ro, windows.ro, hotmail.ro (11/28/2012), google.com.om (4/21/2013), google.ps (8/26/2013), nytimes.com (8/27/2013), leaseweb.com (10/5/2013), google.com.my (10/10/2013), google.cr, yahoo.cr, ebay.co.cr, youtube.co.cr, yahoo.co.cr, flickr.co.cr, amazon.co.cr, msn.co.cr (10/13/2013), eccouncil.org (2/22/2014), craigslist.org, craigslist.com (11/24/2014), lenovo.com (2/25/2015), google.com.vn (2/23/2015), google.com.br (1/3/2017), wikileaks.org (8/31/2017), linux.org (12/7/2018), escrow.com (3/31/2020)
Banks and Bitcoin	stlouisfed.org (4/24/2015), blockchain.info (10/12/2016), blackwallet.co (1/13/2018), wavesplatform.com (7/24/2018)
Credential or Information Stealing	shish.gov.al, mfa.gov.eg, apc.gov.ae, mgov.ae, mea.com.lb, meacorp.com.lb, nsa.gov.iq, dgca.gov.kw, mea.aero, petroleum.gov.eg, e-albania.al, embassy.ly, adpolice.gov.ae, cyta.com.cy, mod.gov.eg, mail.gov.ae, gid.gov.jo, owa.gov.cy, mofa.gov.ae, asp.gov.al, finance.gov.lb (2017-2019) fox-it.com, (9/19/2017)
Malware and Spam Distribution	scrt.ch (7/7/2017), crunchyroll.com (11/4/2017), Angler domains*, Spammy Bear domains*

* Angler and Spammy Bear both involved attacks on thousands of domains, which cannot all be shown here.

APPENDIX A
HIJACKED DOMAINS

The domains hijacked in the known domain hijackings we observed are shown in Table VI. Note that Angler and Spammy Bear were both campaigns that involved thousands of domains.

APPENDIX B
PDNS DATA AND RETRIEVAL

A. PDNS Dataset

The Farsight DNSDB comprises DNS data contributed from sensors located around the world, and from zone transfer files. The dataset has been in construction since 2010 [21]. A report summarizing the dataset at the end of 2019, noted that the DNSDB contained over 130 billion unique RRsets with data for more than 51 billion FQDNs [43]. Farsight filters data to remove responses associated with cache poisoning [42]. The maximum number of RRsets that can be retrieved for a specific query to the database is four million. Farsight provides free, limited access to its DNSDB, as well as research grants. Given the details of domains we used and our methods of retrieval, other researchers should be able recreate the same dataset themselves and reproduce our results.

B. Initial Measurements

For the measurements discussed in Section III, we retrieved records for known hijacked domains that were captured within a time window specific to the attack. For attacks with a specific date given in reports, we limited queries to records with a *time_first* no later than one year after, and *time_last* no earlier than one year before the date. For attacks spanning multiple years, we queried for records with a *time_last* no earlier than the first day of the first year, and *time_first* no later than the last date of the last year. Finally, for attacks occurring within a single year but without specific dates given, we used the year of the attack to set the date range, making the *time_last* no earlier than June of the preceding year and *time_first* no later than June of the following year. Figure 7 illustrates this approach.

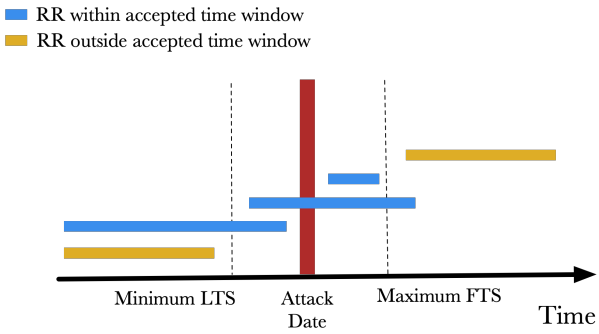


Fig. 7: Records retrieved from Farsight include those captured within a given time window surrounding known attacks.

C. Experiment

In the experiment, we expanded the time frame for which we collected data for the hijacked domains to consider all available data. We also retrieved PDNS records for several popular domains. The latter have a great deal of PDNS data available. As expected, for some of these domains we were not able to retrieve all data within the DNSDB. For those interested in recreating the dataset we used, more information on which domains were in this group, *etc*., can be found in [25].