UNIVERSITY OF CALIFORNIA, SAN DIEGO

# Investigating DNS Hijacking Through High Frequency Measurements

A Thesis submitted in partial satisfaction of the
requirements for the degree of Master of Science

in

Computer Science

by

Benjamin Braun

Committee in charge:

Stefan Savage, Chair
Kirill I. Levchenko
Geoffrey M. Voelker

2016

The Thesis of Benjamin Braun is approved and is acceptable in quality and form for publication on microfilm and electronically:

---

---

---

Chair

University of California, San Diego

2016

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my co-advisors Stefan Savage and Geoffrey Voelker for their excellent guidance, support, and encouragement of my thesis and related research. I would also like to thank my thesis committee member Kirill Levchenko for his insights and support on this project.

Furthermore, I would like to thank Brown Farinholt, who worked closely with me on this project and helped me design and implement parts of the infrastructure required for this project.

I would also like to thank the members of the "Center for Evidence-based Security Research" and the systems and networking group at University of California, San Diego with whom I have interacted during the course of my graduate studies. In particular, I would like to acknowledge Tristan Halvorson for his support with the Spark cluster and the web crawling infrastructure, Neha Chachra for her advice on crawling techniques, He Liu for his help with the existing DNS crawler, and Joe DeBlasio for his assistance on clustering the websites. Thanks to Arjun Roy, Zhaomo Yang and Guo Li - my office mates, for being ever helpful and entertaining.

A special thanks to my family who supported me throughout this journey, my friends, who have always been there to help and encourage me, and finally, Dmitriy Kunitskiy and Johanna Ehlers, who patiently helped me revise my thesis.

ABSTRACT OF THE THESIS

Investigating DNS Hijacking Through High Frequency Measurements

by

Benjamin Braun

Master of Science in Computer Science

University of California, San Diego, 2016

Stefan Savage, Chair

Targeted security threats from resourceful adversaries have become a constant phenomenon on the Internet. One particularly effective attack vector is the Domain Name System (DNS). By compromising the DNS registrar, an attacker can manipulate arbitrary name records of the victim company, resulting in potential compromise of all incoming and internal emails, allowing for highly targeted phishing of login credentials, and a number of other attacks. This thesis examines the prevalence of such DNS hijackings through active scanning measurements of potentially targeted domains and companies.

As part of this work, we implemented and deployed a scanning infrastructure that queries domain name records of a large set of potential targets at high frequency. For further analysis, we also run scans of Transport Layer Security (TLS) certificates, as well as full website crawls when changes are detected.

Over a period of three months, this system collected measurements for 58,000 aerospace related domains. 86% of the scanned domains were stable over the entire measurement period and a majority of the observed DNS changes were caused by content delivery networks and load balancing. We searched this data for attacks using heuristics based on previous DNS hijacking attacks. Although, given our observations, we have not been able to detect ongoing attacks so far, we did observe some anomalies and unspecified behavior. The analysis also showed that short-lived changes occur frequently and we attempt to categorize these by potential causes. Finally, we discuss further improvements to better detect attacks in the future.

# Chapter 1

# Introduction

The domain name system (DNS), which maps human recognizable domain names to IP addresses, is one of the most critical parts of the Internet and required for almost every service, including the web and email. Consequently, the compromise of the associated domain name records of an organization is among the most severe attack vectors. After the focus on DNS cache poisoning attack in 2008 and previous years, the most prominent being the Kaminsky attack [24], the number of known compromises of the DNS has been relatively low compared to other attacks, but each instance had significant impact on the organization affected.

However, in recent years a number of high-profile breaches were publicized, in which weaknesses in the DNS registrar or DNS service provider were exploited. One of the most noticeable incidents was the defacement of the New York Times and Twitter by the Syrian Electronic Army [29], which managed to compromise login credentials for their registrar *Melbourne IT* and redirect their websites for publicity. Although these were most prominent, there is evidence that the compromise of DNS registrars has also been used by nation-state sponsored attackers for covert and highly targeted attacks. One of these attacks used manipulated DNS address records to serve customized malware and exploit users of a remote login portal for a large aerospace manufacturer and collect login credentials.

While there is previous work on performing active measurements to detect DNS cache poisoning, to our knowledge no other group has done this for DNS hijacking through compromised DNS registrars. The objective of this thesis therefore has been to investigate and analyze the prevalence of this attack type by using data collected through high frequency DNS resolutions for a set of aerospace related domains, in particular ones for remote login websites.

To achieve this, we have adapted a scalable DNS crawler that can perform up to 100,000 DNS iterative resolutions every minute from a previous project and record all received records in a database. The set of domains to be monitored is generated using a list of all registered `.aero` domains from the `.aero` DNS zone file and domain names extracted from TLS certificates to guarantee that the domain names are both security-relevant and related to the aerospace sector. To further classify the detected changes, we query and parse TLS certificates for domains with changed address records and perform a crawl of the rendered website.

The remainder of this thesis is organized into five chapters as follows. Chapter 2 introduces the general concepts and necessary details of the DNS required for the understanding of the following work. In Chapter 3 previous DNS hijacking attacks, including both simple defacement attacks and a sophisticated one by resourceful attacker, are presented. In Chapter 4 the design and implementation of the measurement crawler and data processing are described. The collected data is then analyzed and evaluated in Chapter 5. Finally, Chapter 6 completes this thesis with a conclusion and gives an outlook and discussion of potential areas for further work.

# Chapter 2

# Background and related work

This chapter introduces fundamental concepts and terminology used throughout this thesis. The first section presents the technical principles of the Domain Name System (DNS), including an analysis of its security and protection mechanisms. In the following a short overview of Transport Layer Security (TLS) and Advanced Persistent Threats (APT) is provided. Finally, the current literature on DNS hijacking attacks and protection mechanisms is reviewed.

## 2.1 Overview of the Domain Name System (DNS)

The domain name system is a hierarchical, distributed database of resource records (RR) to assign IP addresses to domain names. Initially introduced in 1987 [37, 38], the DNS replaced the single hosts file that was distributed over FTP when its size became unmanageable due to the rapid growth of the Internet.

The primary components of the DNS are the domain name space, name servers, resource records, and resolvers, which will be explained below. The domain name space, structured in a tree form, is used to address records and delegate responsibility for subtrees. Both nodes and leaves of the name space tree contain information stored in resource records. Each node can delegate child nodes to an

independent subzone or manage them itself. Parts of the tree managed by the same authority are called zones of authority.

Each zone designates an authoritative name server which is responsible for hosting the primary records for that zone, containing the original data not obtained through other DNS queries. This information is stored and loaded from the zone file.

The DNS information is queried using DNS resolvers by specifying domain name and record type. There are two main types of queries: An iterative query starts at the root zone and iteratively follows the delegation chain down the tree to the subzone which contains the desired record. In contrast, a recursive query only contacts a single DNS server, a so-called recursive name server, which will perform the iterative resolution as a client itself and then return the final result in a single answer. To reduce load and improve performance, recursive name servers will cache all records according to their time-to-live value.

### 2.1.1 Resource record sets

The data contained in the DNS is stored in resource records (RR). This section introduces the most commonly used types of RRs that will be relevant for the DNS hijacking attacks investigated in this thesis. A full list of all RRs can be found in the original Request for Comments [37, 38] and its updated revisions.

**A record:** Contains a host's IPv4 address.

**AAAA record** Contains a host's IPv6 address.

**CNAME record:** Canonical name mapping an alias to another name.

**MX record:** Mail exchange server receiving emails for the zone.

**NS record:** Name server to use for the zone.

**SOA record** Start of authority contains authoritative information of the zone, the primary name server, zone refresh timings, and administrator contact.

NS records for name servers only contain names and not IP addresses. This can cause a circular dependency when a resolver would have to contact the name server itself for its own IP address. To resolve this dependency, the parent name server will also provide at least one IP address for the name server along with the NS record. This is referred to as a *glue record*.

Each RR has an associated time-to-live (TTL) that decides for how long a record can be cached on a recursive resolver.

### 2.1.2 Further complexities and extensions to the DNS

While sufficient for understanding the following work, the given description of the DNS is in many points a simplification of the full complexity of today's DNS. Over the years, a number of extensions have been implemented to address issues in the initial design, such as adding signatures to records or allowing non-ASCII characters in domain names. An extensive account can be found in [45].

However, the DNS is also used in unintended ways as outlined in [46]. Particularly problematic and widespread abuse of DNS best practices include the use of DNS as a mapping service by content delivery networks and NXDOMAIN remapping. With NXDOMAIN remapping, a DNS server will reply to queries for any subdomain of a domain, even when they do not exist. This is commonly done by Internet service providers or domain parking services to generate ad revenue.

**DNS-based load balancing**

The DNS is frequently used to distribute load among multiple, typically geographically distributed servers. The name server uses the source IP address of the query to reply with an IP address of a server that is either geographically close to the user to reduce round-trip delays or less loaded than other servers. This causes replies to change frequently and requires short TTLs to work effectively, making caching less useful.

Some web services use DNS as a simple means to balance load among servers [8]. The simplest implementation replies in a round-robin way, looping over the pool of available A records. Other adhere closer to best practices and return a randomly permuted list of IP addresses of servers that can be used.

A further problem for measurements is the use of NXDOMAIN remapping. This makes it significantly more challenging to evaluate which domain names actually exists and which replies are the result of remapping.

### 2.1.3 DNS security and compromise

In this section, historical and current attacks on the DNS are presented, as well as examples on how these attacks can be used as further leverage vectors to gain access to privileged internal systems.

When the DNS was initially introduced, primary concerns included fault tolerance, scalability, and flexibility. Security was not mentioned or addressed in the initial specification. Therefore, it is not surprising that a number of security issues with the DNS have surfaced since then. Attacks against the DNS generally fall into one of three categories: denial of service attacks, data modification attacks, and attacks against a user's privacy. In this thesis, we will focus on data modification attacks.

Modification attacks include any unauthorized changes to authoritative records of a zone. Hijacked DNS records can be used for a number of attacks, such as for phishing, as an infection vector for further exploitation, for man-in-the-middle attacks, or simply to generate ad revenue.

**DNS request redirection**

In DNS request redirection attacks, DNS requests are intercepted or redirected to a malicious DNS server that spoofs replies. Common vectors for this attack are the following:

- Vulnerable home routers allow changes to the DNS resolver.
- Malware replaces the DNS configuration on an infected host.
- Malware adds entries to the hosts file on an infected host.
- DNS server operators hijack DNS traffic to generate ad revenue or for censorship [6, 49].

**DNS cache poisoning**

Cache poisoning has been by far the most prominent attack on the DNS. DNS cache poisoning works by forging a reply to a DNS request that is then kept in the cache, poisoning further requests to the DNS server. Already a dangerous attack before Kaminsky showed in 2008 [24], how this attack can be made even more devastating by performing the poisoning for top level domains and NS records instead of single IP addresses. A detailed account on possible attacks can be found in [42].

### DNS registrar compromise

Another way the DNS is frequently compromised is through the domain name registrar. Registrars are responsible for allocating and reserving domain names. Most registrars also provide DNS hosting for their customers. In this case, the registrar can not only control the NS records of a domain, but also any individual A, CNAME, or MX records.

This provides a number of possible further opportunities for the attacker once he has gained access to the DNS registrar. One could simply redirect the home page or other websites within the zone by manipulating the A record, either for the purpose of phishing, infecting clients with malware, denial of service, or defacement.

By taking over the NS record and intercepting any DNS requests for the zone, an attacker can keep the compromise covert and only hijack requests from certain targets. Furthermore, the MX record allows interception of all emails destined for any address within the zone. Controlling A records not only allows for redirection of the website, but also for issuing new trusted TLS certificates with certificate authorities that only require control over the domain name or the administrative email for the domain.

The following lists the different ways of gaining access to a DNS registrar:

- **Social engineering**

  Probably the most common attack vector for DNS hijacking is the use of personal information, either from WHOIS registration data or other sources, to impersonate the account owner and convince the registrar to provide access to the account. Protection against this case is particularly difficult, as it requires the registrar to carefully train its customer support staff and strict

adherence to security policies. There are numerous reported examples of this type of attack.

- **Vulnerabilities in the management web application**

  DNS registrars need to allow users to change their DNS configuration, which is usually done through a web interface. These interface for DNS administration are frequently vulnerable to attacks that allow the manipulation of arbitrary records to take over a domain.

- **Compromise of user credentials**

  If the attacker manages to compromise the access credentials for the DNS registrar, he can hijack all domains controlled by the account. Typical ways to achieve this are phishing attacks, keystroke logging, or other malware infections against the administrator of the organization.

- **Other attacks**

  Other attack vectors include the compromise of the registrar's infrastructure, including the primary name server or databases used to generate the zone file. In other cases, mistakes by the administrators of a company resulted in the domain not being renewed and resold.

### 2.1.4 Defenses to improve DNS security

Given the severity of attacks, a number of defenses have been proposed and implemented to secure the DNS.

**Short term defenses**

When the Kaminsky cache poisoning attack was published, all DNS resolver software vendors implemented protections to increase entropy in the fields of a

query that need to be guessed correctly for an attack as quick solutions. These include the following: randomizing the source port, randomizing the query ID, and randomizing cases of letters in domain, as the DNS is case insensitive. The combination of these made guessing the correct packet content unfeasible to this day. However, this is far from a perfect solution, but rather a fix that could be deployed quickly in light of the critical vulnerability.

**DNSSEC**

Domain Name Security Extension (DNSSEC) is the missing security extension to the DNS introduced in [3, 5, 4]. It adds a hierarchical signature scheme to the DNS to guarantee authenticity and integrity of the provided resolution, while preserving caching and distribution. Though most servers support DNSSEC, only few resolvers actually validate signatures and perform DNSSEC checks, partly because misconfigured deployments cause failures for users [32].

However, when deployed correctly, DNSSEC provides an effective protection against cache poisoning and spoofing attacks. Nevertheless, in the case of a compromised DNS registrar, DNSSEC is likely not going to be useful at this point. While obtaining signatures for changed records is not possible without access to the zone signing key or key signing key, the signatures can simply be removed from the zone without causing too much suspicion due to the limited deployment. In the future, when DNSSEC is fully enforced, it should make DNS hijacking significantly harder, even when controlling the DNS registrar.

**Registrar protections**

Most DNS registrars implement additional protection services for domains. These commonly include protection against unauthorized domain transfers, automatic domain renewal, and private registration to reduce the risks of social

engineering.

## 2.2 Transport Layer Security (TLS)

Transport Layer Security (TLS) is the de-facto standard for point-to-point communication on the Internet, used widely for, among others, securing websites, email transfer, or virtual private networks. Standardized in [17], TLS and its predecessor, Secure Sockets Layer (SSL), have undergone significant changes to the protocol to counteract security vulnerabilities.

Among other information, certificates for web servers need to contain the domain name to guarantee authenticity. This is stored either in the common name or subject alternative name (SAN) field when using the extension, allowing for multiple domains in a single certificate. An analysis of the certificate ecosystem based on Internet-wide scans on the default HTTPS TCP port 443 is presented by Durumeric et al. in [20].

## 2.3 Advanced Persistent Threats (APT)

Target cyber attack against individuals and organizations by highly sophisticated actors present an increasing threat. According to the definition by the U.S. National Institute of Standards and Technology in [40], an APT has the following characteristics that distinguish it from traditional adversaries:

> "An adversary that possesses sophisticated levels of expertise and significant resources which allow it to create opportunities to achieve its objectives by using multiple attack vectors (e.g., cyber, physical, and deception). These objectives typically include establishing and extending footholds within the information technology infrastructure of the targeted organizations for purposes of exfiltrating information, undermining or impeding critical aspects of a mission, program, or organization; or positioning itself to carry out these objectives in the

> future. The advanced persistent threat: (i) pursues its objectives repeatedly over an extended period of time; (ii) adapts to defenders' efforts to resist it; and (iii) is determined to maintain the level of interaction needed to execute its objectives."

Similarly in [44], the more general term of targeted attacks is characterized by low volume attacks against a clearly preselected target with sophisticated malware, often using 0-day exploits, which are exploits using previously unknown security vulnerabilities, for the initial compromise. Typical examples of APTs are nation state attackers, terrorist groups, or state-sponsored groups.

However, a recent trend noted by researchers from FireEye [34] is that the gap between the skills and tools of traditional attackers and advanced persistent threats are closing. Traditional attackers have adapted similar techniques previously exclusive to nation-state sponsored attackers and nation-state attackers are increasingly using commonly available exploitation toolkits and products to cover their tracks.

Based on a corpus of 26,000 target email attacks from Symantec [44], the aerospace and defence industry were both highly targeted by advanced threats in the recent past.

A detailed analysis of the behavior and methods of advanced persistent threats can be found in [11].

## 2.4 Related work

This section offers an overview of related studies that have analyzed DNS security in the past. While previous work on active measurement studies to detect compromised DNS servers is relatively sparse, there is a large body of work on security measures and analyses to improve DNS security.

### 2.4.1 Active DNS security measurement studies

A study by Wessels from 2006 [48] searched for and investigated DNS cache poisoning. In the investigated attacks, vulnerable caching resolvers trust out-of-bailiwick referrals for parent name servers and store them in their cache, something that even at the time most resolvers were not vulnerable to. For the study, Wessels et al. actively queried authoritative names servers from a large set of domain names and zones to find authoritative name servers that performed parent-zone poisoning. While they were able to find some examples in the wild, many of them appeared to be the result of administration errors rather than malicious intent. A large fraction of the domains resulting in cache poisoning were either parked or expired. In most cases, the poisoning resulted in redirects to advertising sites. However, even in 2006, most resolvers were not vulnerable to this attack as they implemented a bailiwick policy to prevent them.

Anax by Antonakakis et al. [2] is an active monitoring system to detect DNS cache poisoning in open resolvers using Internet-wide DNS measurements and machine learning. The system relies on the fact that the DNS consists of "known, usually stable set of NS records" [2], while in the case of attacks new IP addresses need to be introduced. For their measurements, they query a diverse set of 300,000 open recursive DNS servers for 131 domains of interest. From these measurements they generated ground truth data using a set of rules to distinguish between legitimate responses, content delivery networks (CDN), misconfigurations, and poisoning. Their primary rule for this was that the resolved IP address needs to be associated with the organization of the domain or a CDN. They used the CIDR registration database to generate these associations, although they also manually labeled a significant portion of the IP addresses. With this ground truth data they

trained a classifier that they claimed to achieve a false positive rate of 0.6% with a true positive rate of 91.9%.

However, their system is limited in a number of ways: The most significant issue is that they do not further elaborate on the detected cases of cache poisoning. While they assure that these are hand-verified, their criteria for poisoning are questionable: any IP address that appears on a blacklist and is not associated with the domain owner. However, according to [41], IP blacklists are far from accurate, containing non-trivial amounts of both false positives and false negatives. It seems likely that not all identified cases were caused by malicious intent. The second limitation is the small set of domains they investigated. While this might make it easier to determine IP to organization mappings, it ignores other high value targets of DNS poisoning. Finally, they do not specify the frequency with which they are querying the DNS resolvers and therefore might miss short lived poisoning attacks.

In contrast to this, we focus on compromised DNS registrars, as cache poisoning has become more complicated due to additional query entropy and other security measures. Instead of investigating the Alexa top domains and top e-business websites, we focus on the aerospace sector, a sector frequently targeted by advanced persistent threats and nation-state attackers. Furthermore, our goal was to detect attacks lasting only minutes.

# Chapter 3

# Recent DNS registrar compromises

In this section, we will analyze and characterize known attacks from the recent past that have used DNS hijacking through compromised DNS registrars. In the first section, attacks against organizations consisting of website defacement for the purpose of publicity or activism are outlined. In the following sections we will focus on more covert, sophisticated attacks assumed to be performed by nation state sponsored groups or other advanced persistent threats.

## 3.1 Website defacements using DNS hijacking

There are several examples of attacks that used DNS hijacking through a compromised DNS registrar to redirect popular websites. In all of the following examples, the original websites were redirected to defacement sites, which contained political statements in some of the cases. Given the instantly noticeable effects of such an attack, this kind of attack usually does not last for more than a short period of time, depending on the configured time to live of the records affected by it.

Recent examples of such attacks are numerous: The New York Times and

Twitter in August 2013 [29], LeaseWeb [14], Rapid7, and Metasploit in October 2013 [36], eBay in February 2014 [31], Craigslist in November 2014 [9], Lenovo and Google Vietnam in February 2015 [26], and the University of Connecticut in December 2015 [13]. In all these cases, while being quite disruptive for the affected website, the attacks were mitigated within hours and had no lasting impacts.

## 3.2 French aerospace companies targeted by DNS hijacking

A particularly interesting case of an attack using compromised DNS is the attack against French aerospace companies in the beginning of 2014. In February 2014 remote users of Snecma S.A., a French aircraft and rocket engine manufacturer and subsidiary of Safran S.A. were targeted and compromised, as originally reported by researchers from Seculert [39].

The technical details of the attack indicate a highly sophisticated attack. The attacker involved long-term planning and commitment of significant development resources. According to the analysis published by Crowdstrike [15] and an FBI alert [18], the attackers presumably acted as follows:

- At some point before the attack, the hackers managed to compromise the DNS registrar used by Snecma. They then diverted the NS records for the `snecma.fr.` zone to name servers under their control.

- For short periods of a few minutes, the malicious DNS server answered A record queries for remote login pages with an IP address controlled by the attackers, redirecting employees and other clients to the malicious website. The malicious host's IP address was pointed to by multiple domain names similar to existing domain names used by Snecma.

- The malicious host served a website containing malware which exploited a use-after-free 0-day vulnerability in Microsoft Internet Explorer 10 [1].

- After infecting a user's machine, the malware added entries to the hosts file of the client. Somewhat surprisingly, these entries were for domains which provide remote access to internal networks for employees, but also other suppliers and partnering aerospace companies and pointed to the correct hosts inside the company. The purpose of these entries seemed to be guaranteeing connectivity to the correct hosts even during an ongoing DNS redirection.

- When the employee or authorized user tried to login to the remote access site, the malware collected their credentials using a keylogger and sent them back to the command and control server of the attacker, the same host that was used to serve the malware.

As the targeted remote access portals of Snecma are used by a significant number of suppliers and partners, the attack can be considered a watering hole attack clearly targeted against the aerospace sector. However, the extent of the compromise remains unclear.

### 3.2.1 Related attacks by similar actors

A previous attack against microturbine manufacturer Capstone Turbine Corporation in 2012 shared some commonalities [15]: In both cases, Internet Explorer 0-days were used to infect clients and a remote access trojan from the Sakula malware family was used. Additionally, in both attacks domain names related to the French Aerospace Industries Association were used for hosting the malware.

The 0-day exploit used in the attack against Snecma was also used against other organizations in the same time frame. One of these is the compromise of the U.S. Veterans of Foreign Wars' website by an adversary named Aurora Panda by CrowdStrike. The website was compromised in an unknown manner and served the same malware exploiting the same 0-day vulnerability in IE 10 as in the Snecma attack [25]. However, the remote access trojans deployed afterwards were of different types.

## 3.3 U.S. Office of Personnel Management data breach

In a major data breach, hackers managed to exfiltrate personally identifiable information, including personnel records and fingerprints from the U.S. Office of Personnel Management (OPM) between May 2014 and April 2015 [35]. The data contained 22.1 million records ranging from Social Security numbers, names, addresses, and date of birth of government employees to potentially information on security clearances and data from background checks [22].

According to the investigators' timeline [33], the initial breach of the internal networks occurred on May 7th 2014. The adversaries apparently used stolen credentials of employees to login to machines. In an advisory notification by the FBI cyber task force to private industry [18], the FBI highlights that the group responsible for the breach of the OPM poses the capability of "DNS hijacking facilitated through the compromise of DNS registrars". The group has been observed in using this technique but it is unclear from the alert what role it played in the breach of the OPM.

After gaining initial access, the group planted remote access trojans (RAT) on internal machines, including variants of the Sakula RAT, FF Rat, and Trojan.BLT

RAT. They then moved laterally trough the internal network, in order to reach data centers containing more sensitive personal records. It took until April 2015 for the breach to be detected and mitigated, when suspicious TLS encrypted traffic leaving the data center raised alerts.

The group behind the attacks is suspected to be originating from China [22]. Indications of a state sponsored attack include the fact that no data from the breach leaked to underground forum or other markets to be sold, making espionage the more plausible objective. However, pinpointing an attack to a specific adversary is challenging, as state sponsored, persistent threats often use off-the-shelf remote access tools and exploits to cloak their identity.

According to a report from September 2015 the data stolen from OPM had been used to breach U.S. defense contractors [10]. The data allowed for highly targeted attacks, such as advanced social engineering or spear phishing using the personal information of employees with security clearances.

## 3.4 eNom and St. Louis Federal Reserve DNS breach

On May 18th the St. Louis Federal Reserve notified its users of an attack on April 24, 2015 against their research website [16, 27]. According to the press release, the attackers managed to hijack DNS address records of multiple subdomains and redirected some of the traffic to similar looking websites.

The limited technical details provided in the release suggest that the attackers managed to compromise the DNS registrar used by the Federal Reserve and replace the name server entries to ones controlled by them. They then modified the A records for some DNS requests to redirect clients to a malicious web server, which served a page that closely resembled https://research.stlouisfed.org. It is unclear

whether the page was only phishing for user credentials or if it also served malware.

The affected subdomains where all used to host the Federal Reserve Economic Data and other macroeconomic datasets. This data is not only accessed by banks and other financial institutions but also by researchers, journalists, and policy makers. Given that the data on these websites is openly accessible, it seems likely that the attack was also a watering whole attack against the users of these websites, similar to the attack against Snecma.

In a notification [43] to its customers two days later, the DNS registrar eNom explained that they became the victim of a"very sophisticated attack by a group that targets large internet infrastructure companies". According to the notification, the group hijacked the DNS traffic of 4 domains, but did not mention which ones in particular. Given the timing, it seems likely that the St. Louis Federal Reserve's domain was one of the affected ones.

The attack lasted only for "a very short period of time" and was mitigated within hours. No further user data or other domains had been affected by the breach. Considering how targeted the attack was executed, the objective of the attack, and the required skills and planning, it can be presumed that a state sponsored actor or foreign adversary is responsible for the attack.

## 3.5 Summary

The presented examples of recent attacks show that the compromise of DNS registrars has become a common vector for data breaches against organizations in sectors with high-value information. However, it is difficult to say how prevalent such attacks really are, as there are presumably cases in which the compromise is either never publicized given the sensitive nature of the targets and the involvement of state sponsored attackers or it is not even detected. Except in the obvious cases

outlined in the first section of this chapter, an attack can be conducted in a covert manner.

To better answer these questions, the following parts of this thesis are dedicated to building a measurement system that can detect compromised DNS entries, in particular their use for advanced persistent threats and cyber espionage, and analyze the collected data for signs of an attack.

# Chapter 4

# Data Collection

In order to detect ongoing attacks against high value targets, we first created a lists of domain names from companies that might be the target of these attacks. Then, given this list of relevant domain names, we observe any changes to their DNS entries by crawling them at a high frequency. In this chapter, we introduce how we combined the data from existing scans for TLS certificates and `.aero` zone file data to generate a list aerospace related domains.

Furthermore, we present the design and implementation of a crawler that can resolve up to 100,000 domain names iteratively each minute using a single machine. To allow for further forensic analysis of suspicious incidents, we also built a crawler that retrieves TLS certificates from servers with a detected change of the DNS entries. As a last layer, we also implemented a web crawler to request and render the website and take a screenshot to allow for retroactive analysis of detected attacks.

With this data collection system, we are able to detect even short lived changes to domain names and can use heuristics to distinguish between actual attacks and regular maintenance activity.

## 4.1 Generating a list of relevant domains

The following section explains how domains were selected that could potentially be targeted by attackers. As described in Section 3.2, aerospace has been a primary target in the past. Therefore, we decided to focus on this industry first.

### 4.1.1 TLS certificate data

The first step for this was to extract the domain names from a crawl of the entire IPv4 address space. The ZMap Team at the University of Michigan publishes a weekly dump of fully parsed TLS records and X.509 certificates collected by performing a TCP SYN targeting port 443 across all IPv4 addresses [19]. To generate the list of domains, the scan from October 21 2015 was used. This scan contained almost 48 million leaf certificates deployed by websites.

One limitation of using the certificate data only from a full IPv4 scan is that it will not include certificates from servers using Server Name Indication (SNI) [7]. Whenever a server is using multiple certificates on a single IP address, the data will only contain the default one. However, we expected this to only be a minor limitation. As noted in [20], only very few hosts actually use SNI, and this number could be even lower for servers hosting login pages.

### 4.1.2 Extracting domain names from certificates

From the parsed certificates, we extracted both the common name (CN) and, where available, the subject alternative name (SAN). The certificates contained 8,782,198 unique common names and 11,247,749 unique subject alternative names, with a total of 14,991,941 unique domain names.

To limit the domains to aerospace related ones, we used the `.aero` top-level domain (TLD) to find relevant second-level domains (SLD). As the `.aero` TLD

exclusively allows only companies, organizations, associations, government agencies, and individuals in aerospace and aviation-related fields that pass the screening process to register domains within that TLD, we can utilize this to filter relevant domains.

In a first step, we extracted everything from the CN and SAN fields that was a valid domain name and filtered for the ones that have a SLD that also appeared in the `.aero` zone file. This resulted in a list of domains from organizations that are likely aerospace related. Overall this yielded 62,367 domain names, of which 47,924 actually resolved to an IP address.

From these domain names, we then extracted common third-level domain components. Through manual inspection of the frequently occurring domain prefixes, we selected the ones that seemed most likely to be used for internal or external facing login pages. The resulting selection with frequency counts is shown in Figure 4.1 and a full list of the 100 most frequently occurring prefixes can be found in Figure A.1 in the appendix.

These prefixes were then combined with the second-level domains from the `.aero` zone file and the `.com` top-level domain, generating for the domain name `login.boeing.com` the prefix `login`, the SLD `boeing` and the `.com` TLD. This resulted in a total of 32,7080 generated domain names. Out of these, only 11,887 actually resolved to an IP address.

Figure 4.2 shows the total number of generated and collected domains and respectively the portion of the resolving domain names. The vast majority was extracted from the certificates. A particular problem with the generated domains proved to be NXDOMAIN redirects, which resulted in valid DNS lookups even for non-existing domains. However, all generated domains with a SLD were excluded, if NXDOMAIN redirects were detected for all prefixes.

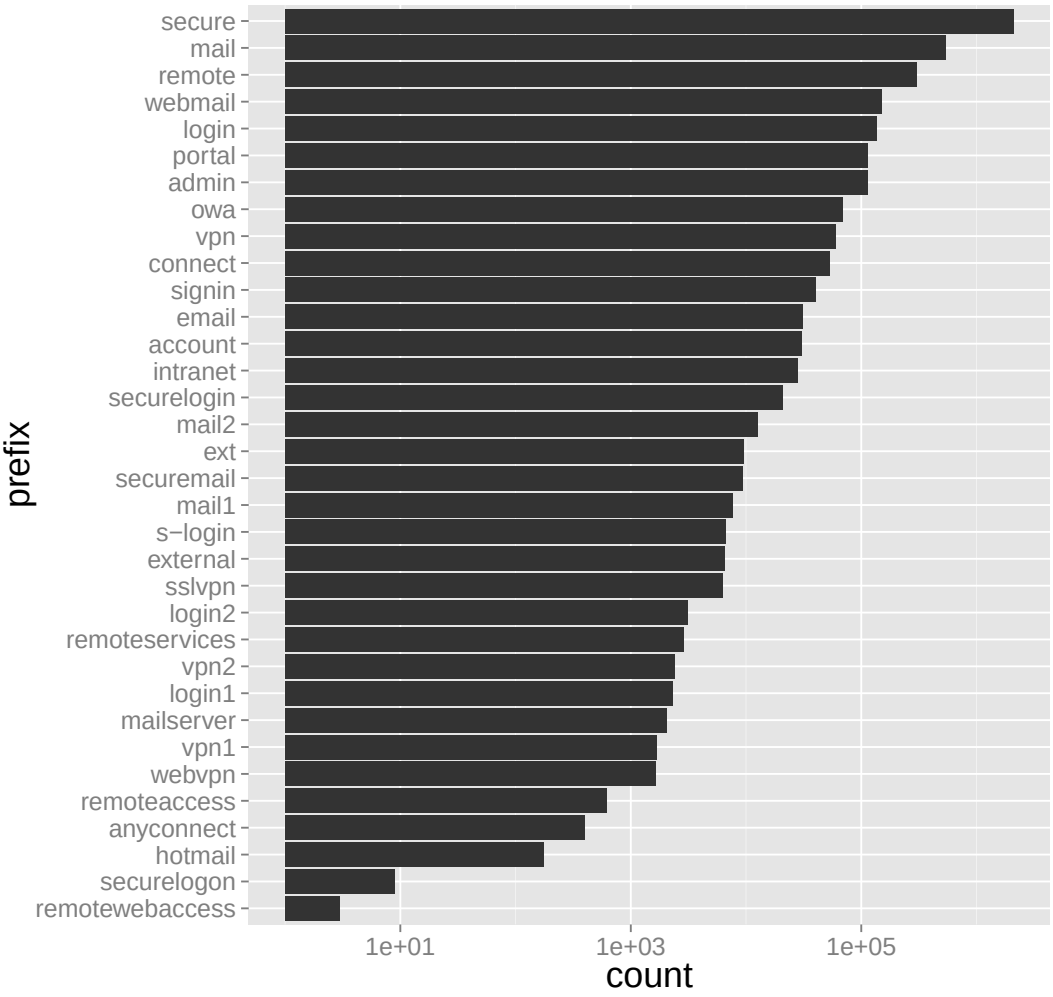


**Figure 4.1.** Top 100 most frequently appearing domain prefixes in TLS certificates

### 4.1.3 Limitations

There are two obvious limitations to the given approach. For one, we might not include domains from organizations that use wildcard certificates. The approach of constructing the domains from the fragments might catch some of these, but most likely not all of them.

The second issue is that there might be too many domains included, in particular ones which are not aerospace related. This might make the collected

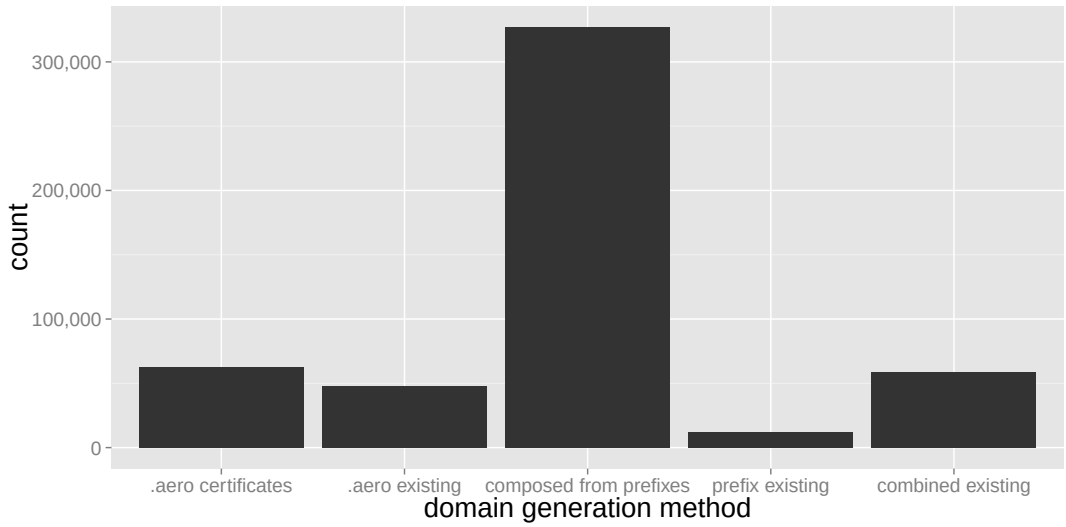


**Figure 4.2.** Overview of collected and generated domains used for scanning

data noisier, but should not influence the results otherwise.

## 4.2 Scanning system architecture

In the following we will give an overview of the scanning architecture, first describing how the overall system was designed and then explaining each component in detail. The general architecture of the scanning system is depicted in Figure 4.3. The system components interact as follows: For both communication and queueing of results and messages between the scanning services, we use a Redis key-value store. The list of domains is stored in a key in Redis and retrieved by the DNS crawler on every scan, allowing for dynamic updates without interruption. The DNS crawler then queries the DNS servers for each domain iteratively through the proxy host. The results of the crawler are entered into a queue in Redis for processing and data deduplication. A Python service retrieves records from this queue, compares them with the latest database entry, and adds the data to the database if any information has changed compared to the previous result.

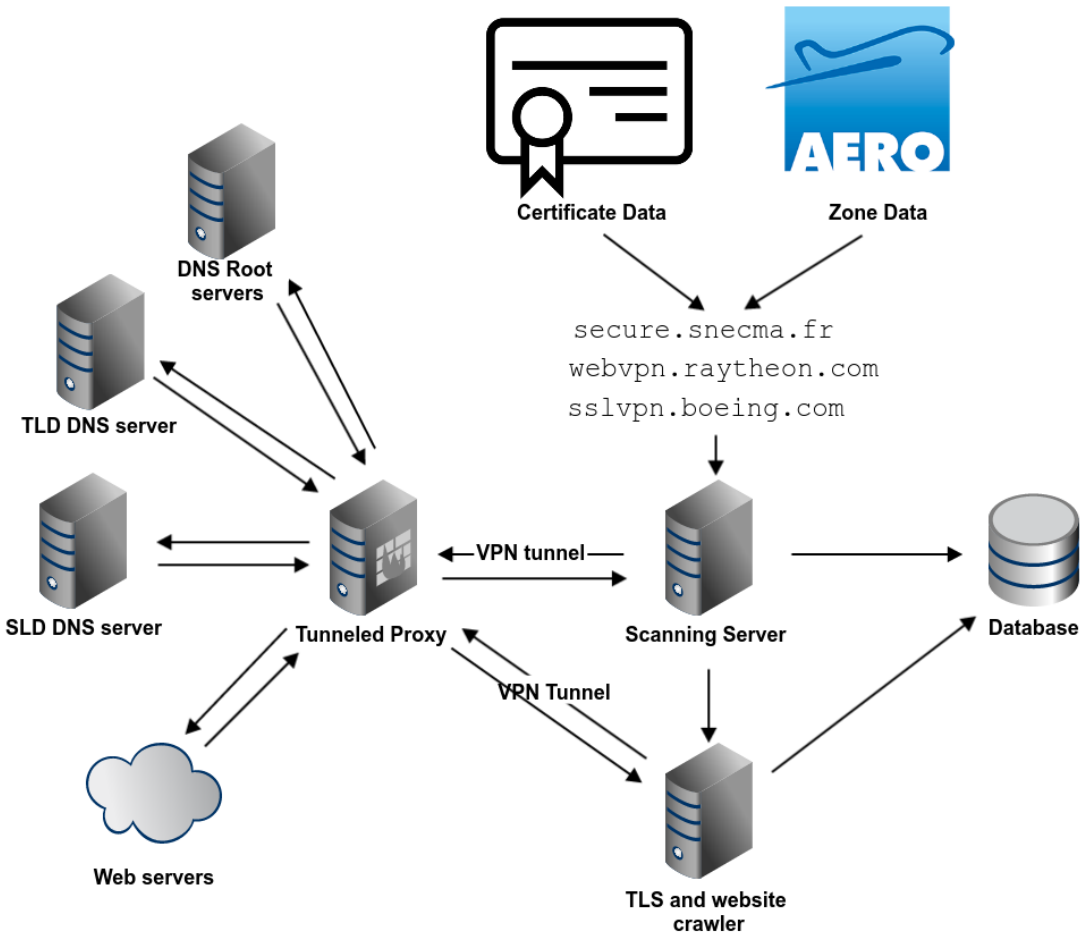


**Figure 4.3.** Overview of the DNS scanning system

Whenever a change in the DNS result is detected, the TLS certificate crawler is notified through Redis. A change is defined as any result that differs in the set of IP addresses returned, the common name chain or the set of name servers. The certificate crawler batches requests for up to 30 seconds and then performs a crawl of all hosts in the batch. Results are again deduplicated before storage; only when a new certificate is observed for a host is a new record in the database is added. As in the previous case, whenever a new certificate is observed, the web crawler is used to render the page and store a screenshot, as well as the content of the DOM for further forensic analysis in the database.

## 4.3 High frequency scanning of DNS

While single DNS requests are quite small and easy to perform, querying around 60,000 domains in short succession is more challenging.

The basis for the DNS crawler was provided by He Liu and has previously been used to resolve DNS chains for the click trajectory work [30]. The dig8 DNS crawler is written in the Go programming language and resolves DNS records in parallel for a large set of domains. The crawler performs a full iterative lookup, starting at the root zone. It resolves CNAMES and returns all A records and authoritative name servers including their IP addresses. To allow for easier post processing, we modified the crawler to write the output to a Redis queue using JSON encoding. The crawler parallelizes outgoing requests to all servers in the batch. The total time required to perform the crawl of a batch is usually limited by the slowest lookup or potential timeouts. This causes the crawls to frequently take longer than the minute between successive crawls. However, the subsequent batch still starts after a minute and runs in parallel. While queries are not occurring precisely every minute, the deviations are small enough not to influence the results.

## 4.4 Preventing scan detection

To avoid issues with DNS response rate limiting [47] and in general with the scans being detected, we implemented a few measures for preventing this. As these scans would be easily detectable given the frequent periodic queries, the DNS crawler uses randomized IP addresses from a total pool of six /24 IP address ranges for each batch of queries. Additionally, caching is enabled to limit the request to intermediate name servers. With this large pool of outgoing IP addresses and the reduced queries to individual name servers, the crawling should not be

triggering typical DNS rate limits that are in place to prevent denial-of-service or DNS amplification attacks.

## 4.5 Result processing

A processing script fetches the DNS crawler results from Redis and parses them for storing in a database. To minimize storage, only differentials are stored in the database. Whenever either the A record or the CNAME chain changes for a domain, a new record is added to the database. Each database record includes a timestamp, the domain, and all resolved DNS data. A full database schema can be found in the appendix in Figure A.2.

## 4.6 Certificate crawling

If the IP address for a domain changes, we could be detecting a potential ongoing attack. To distinguish between intended IP address changes and malicious ones, we request a servers' TLS certificate on a change. To avoid having to parse the certificate ourselves, we use ZGrab to perform the TLS handshake. ZGrab is developed by Durumeric et al. [19] and is typically used together with Zmap [21].

ZGrab supports Server Name Indication (SNI) [7], as well as validity checks of the received certificate. The certificate is parsed into JSON for easier analysis. ZGrab includes a check against the specified browser root store to see if the certificate is trusted, as well as to test whether the given domain name matches the certificate. We have configured ZGrab to validate certificates against the Google Chrome root store. The parsed certificates were stored together with the raw binary data.

To limit redundant queries, a domain and IP address pair is not crawled again after a certificate has been successfully received previously, as we assumed

that DNS hijacking would redirect to a server controlled by an attacker and this seems unlikely for a company internal server. However, if previous requests failed, the IP address is queried again to distinguish between temporary unavailability and a server without support for HTTPS.

## 4.7 Website crawling infrastructure

Initially, all domains were crawled once to provide reference data to compare against in case an attack is observed. For each domain, the fully rendered page source is stored, as well as a screenshot of the page.

With the described preconditions for triggering a website crawl, the number of websites to be crawled is relatively low during continuous operation. Therefore, the web crawling infrastructure was kept simple. We use Selenium to drive a Firefox instance to load the website and take a screenshot of it. This data is stored in a separate database table.

# Chapter 5

# Data analysis and results

In this chapter the results from data collected over three month, starting on December 14th 2015 until March 7th 2016 are presented. While the majority (86%) of domains never changed during this period of time, we recorded a total of over 73 million changes for the remaining domains. However, this number is highly inflated, mostly caused by load balancing and content delivery networks. Through content clustering of the domains' websites, we were able to verify that a significant portion of the crawled domains were indeed login pages for aerospace related companies.

While we could not find definite indications of ongoing DNS hijacking attacks, we were able to identify a number of cases with anomalous behavior and configuration errors leading to unavailability. A comparison of the crawled data with IP blacklists and historic zone data proved to be of limited use due to the large number of false positives and insufficient data to distinguish intended changes from malicious ones. However, with a longer measurement period, the probability of observing DNS hijackings should be increasing.

## 5.1 General data exploration

Over the measurement period from December 14th until March 7th, a total of more than 73M DNS changes were observed. However, this number includes a

significant number of redundancies that will be further described in the following. Hence, only 103,766 distinct IP addresses have been recorded, using only 30,461 distinct certificates.

Most notable is that 86% of domains never changed during the entire period. That means their A, CNAME, and NS records were entirely stable during the measurement period. This confirmed our initial assumption that the DNS contains mostly stable information. For these domains, we can also be certain that no DNS hijacking occurred. Therefore, the following analysis will focus on the remaining 14% of domains with at least one observed change.

As Figure 5.1 shows, out of the domains with changes, over 50% had less than 100 changes during the time period. This means that conversely the other half of the domains were causing nearly all observed changes. For these domains with over 1,000 observed changes, there seem to be two common explanations:

**Content delivery networks (CDN):** Typically, CDNs will provide a list of IP addresses for a requested domain and the client can pick a random IP address from that list to connect to. Depending on the number of available IP addresses, the returned set can change on every or almost every request. This effect is further amplified by the source address randomization. Since consecutive requests originate from different /24 networks, the results of these queries often differ as well.

**Round-robin DNS load balancing:** A number of hosts appear to be using a simple round-robin scheme for load balancing, returning only a single IP address each time. One reason for this behavior is that older operating systems including Microsoft Windows XP and Vista do not pick a random IP address when presented with a list of addresses. Therefore operators limit the DNS

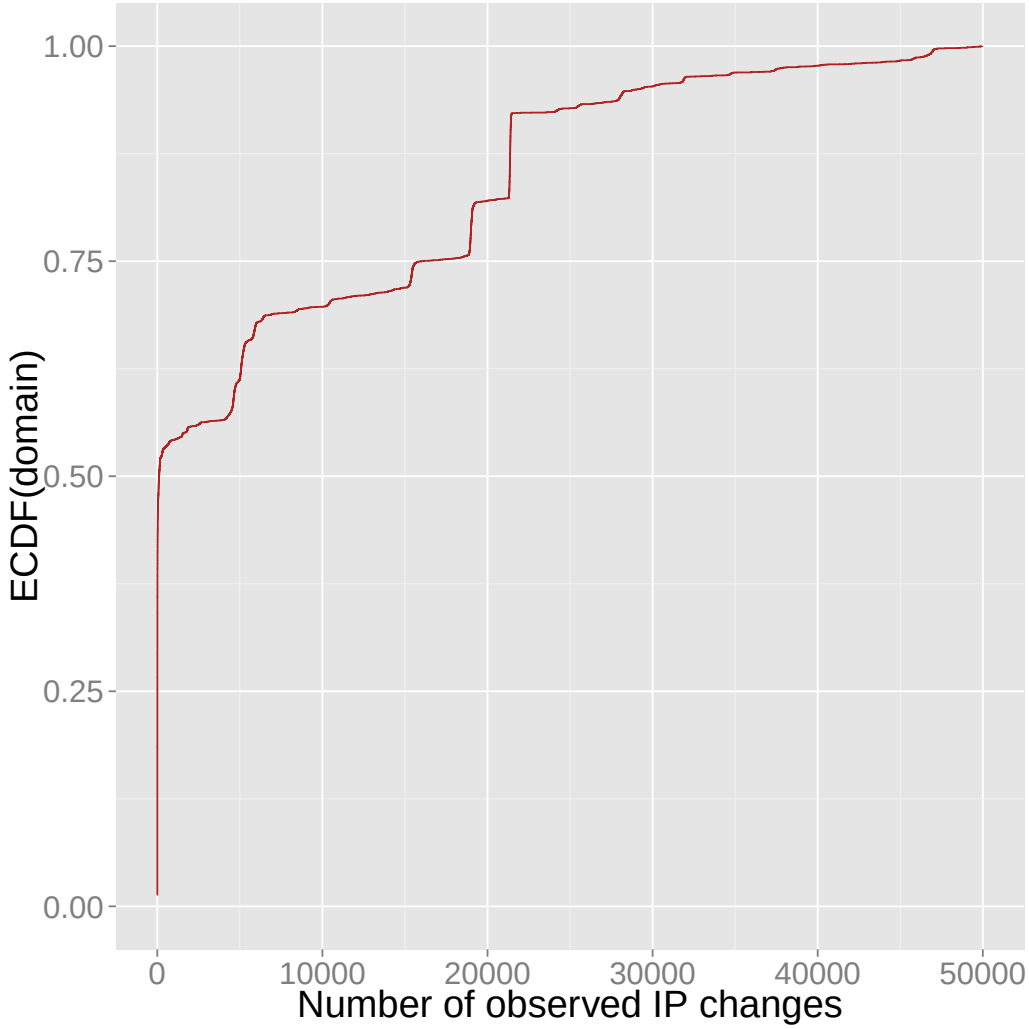


**Figure 5.1.** Empirical cumulative distribution function of the domains over the number of observed IP address changes

response to a single IP address to achieve distribution. Other explanations might be the simplicity of implementing such a system and the availability of plugins supporting this mode for popular name servers. However, this results in a highly inflated number of observed changes by our system, as even with very few available IP addresses, the observed result will change each time.

However, the collected TLS certificate data is particularly useful when analyzing these changes. When the observed certificate is identical across the observed IP addresses, we can assume that these IP addresses are all under the organization's control and not the result of DNS hijacking. The possibility of an attack only needs to be considered when a new certificate or a server with a new IP address appears in the observation.

## 5.2 Domains mapping to identical CNAMEs

From the empirical cumulative distribution of the domains, it is obvious that there are groups of domains behaving almost equally, and causing steps in the graph. These domains with equal patterns seem to be mapping to the same common name. Figure 5.2 shows the number of observed changes aggregated by common names. A single common name, `secureserver.net`, is causing almost a third of all recorded changes. Other large content delivery networks follow with some distance. The large number of domains mapping to the `secureserver.net` common name is the result of how we generated the list of domains. This is the default web mail service automatically provided by the domain registrar GoDaddy. Therefore most parked domains had this service enabled and we had a total of 1,183 domains redirecting to that service.

To reduce the redundant data being stored, only changes to the fully resolved

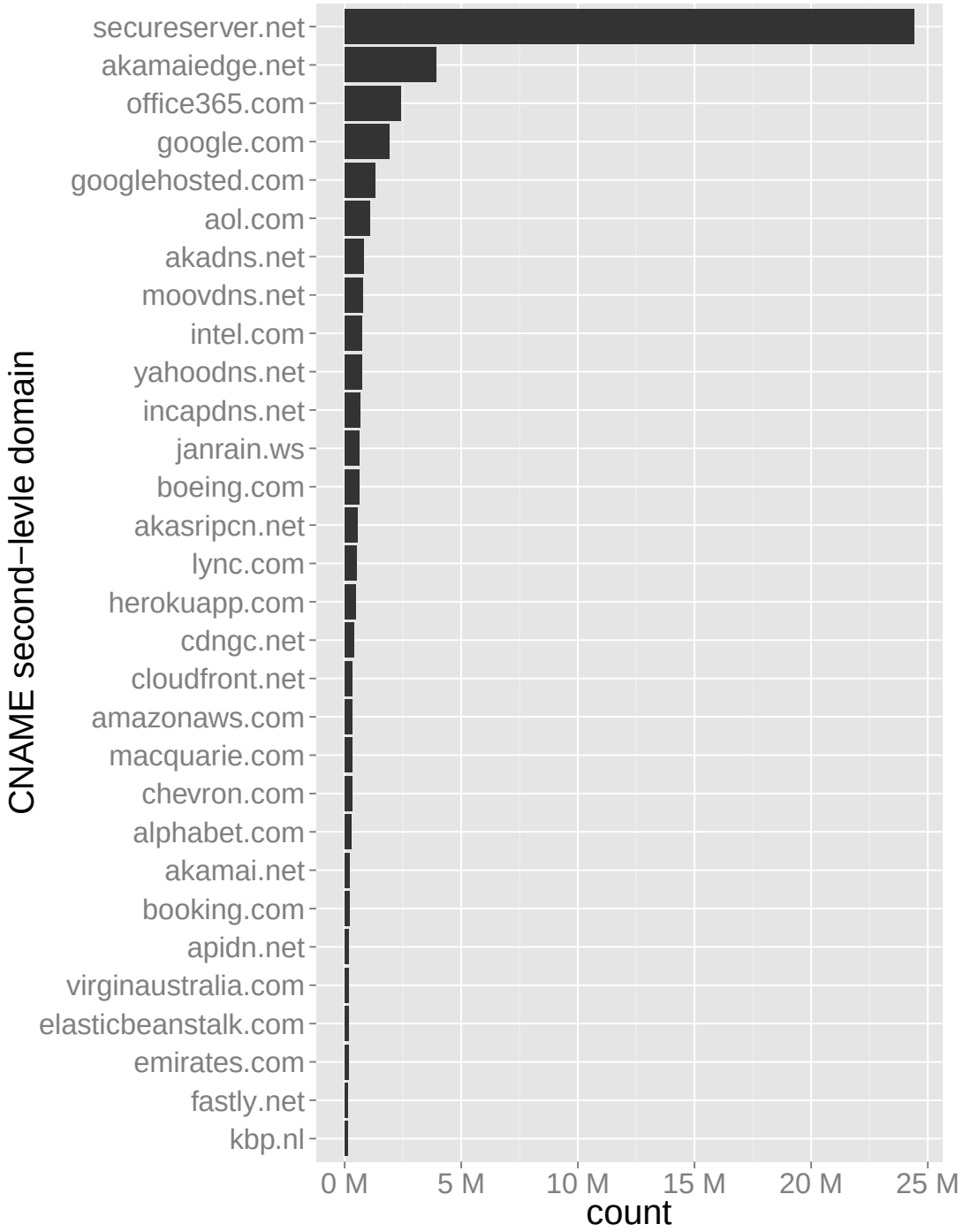


**Figure 5.2.** Count of second-level domains in common names

CNAMES were recorded for the most frequent common names starting in January. Domains resolving to any of these CNAMES were only updated if the CNAME chain changed and otherwise ignored when searching for DNS hijacking attacks.

## 5.3 Classification of observed short-lived DNS changes

In the following, we further characterize the observed DNS changes. Except for the load balancing case, we had assumed that short-lived changes to DNS information are the exception, pointing to attacks or configuration errors. However, when analyzing the median time between observed changes for each domain as shown in Figure 5.3, we found that even for domains with few total observed changes, the common case is a change lasting only a few minutes to hours.

As we can see from the histogram in Figure 5.3, the majority of all changes for domains with a total number of changes above ten lasted only for minutes to a few hours. Only for domains with a total of ten or less changes does the configuration persist for multiple days.

Another pattern observed in the chart is the use of NXDOMAIN redirects. Most of the spikes at specific time points are caused by a single SLD that redirects to a single IP for a number of prefixes, causing identical behavior.

From examining the data, we identified the following typical cases that explain the common occurrence of short-lived A record changes on the timescale of a few minutes:

- **DNS used for failover**

  A record changes to another IP address for few minutes and then reverted back. A potential example of this behavior is shown in Table 5.1 for the

**Figure 5.3.** Median IP change time for all domains. Histogram generated using time bins with a width of 500 minutes.

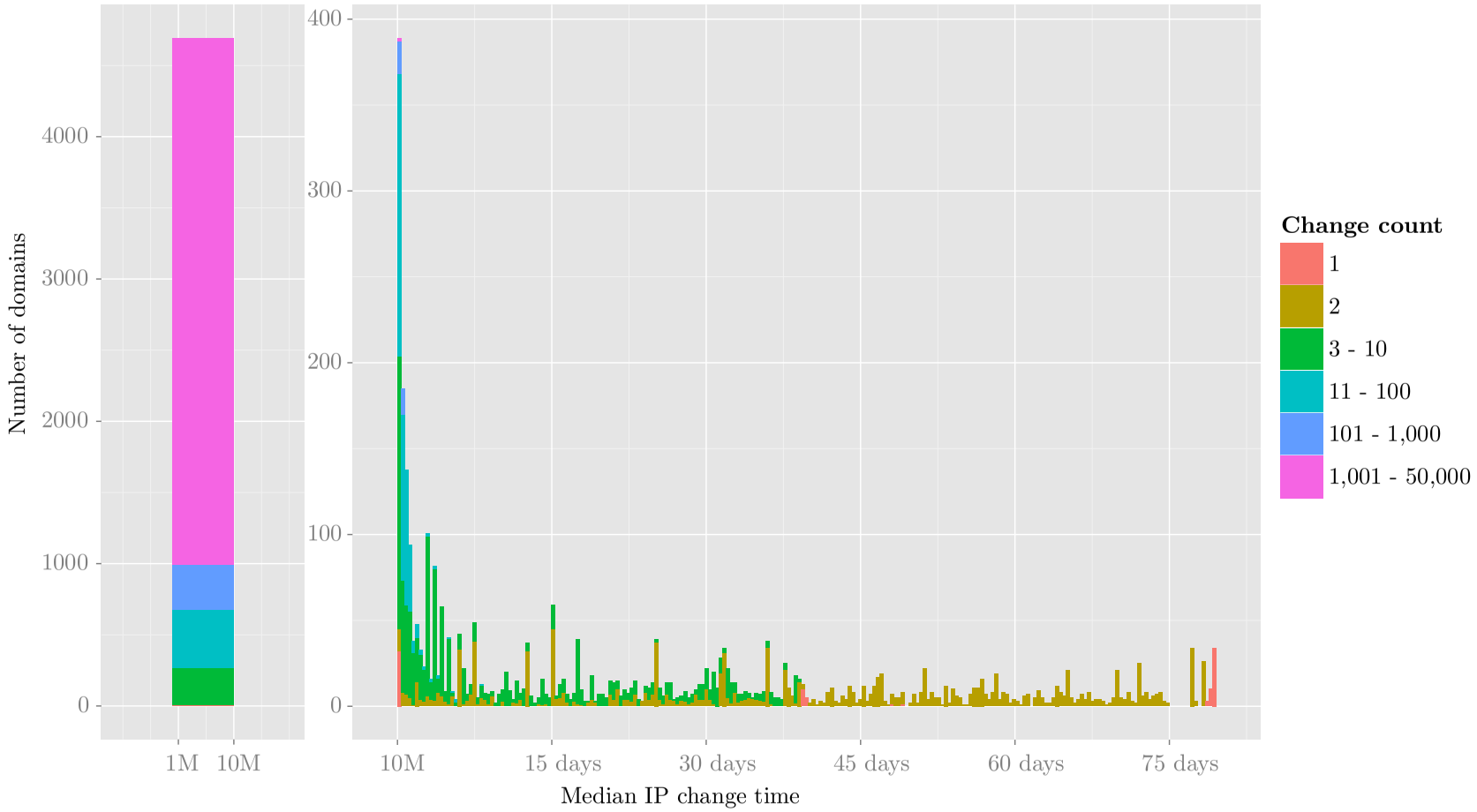
Number of domains
0
1000
2000
3000
4000
1M
10M
400
300
200
100
0
10M
15 days
30 days
45 days
60 days
75 days
Median IP change time
Change count
1
2
3 - 10
11 - 100
101 - 1,000
1,001 - 50,000

**Table 5.1.** DNS resolutions and certificates for webvpn.raytheon.com

| Timestamp | IP addresses | Certificate | Cert ID |
|---|---|---|---|
| 2015-12-14 04:51:57 | 199.46.217.46 | trusted, issued by Entrust | 13110 |
| 2015-12-15 16:05:27 | 199.46.216.43 | trusted, issued by Entrust | 28432 |
| 2015-12-15 16:57:20 | 199.46.217.46 | trusted, issued by Entrust | 13110 |
| 2016-01-04 08:04:35 | 199.46.216.43 | trusted, issued by Entrust | 28432 |
| 2016-01-04 08:05:43 | 199.46.217.46 | trusted, issued by Entrust | 13110 |
| 2016-01-07 04:44:38 | 199.46.216.43 | trusted, issued by Entrust | 28432 |
| 2016-01-15 05:00:40 | 199.46.217.46 | trusted, issued by Entrust | 13110 |
| 2016-03-05 10:07:18 | 199.46.216.43 | trusted, issued by Entrust | 28432 |
| 2016-03-05 10:08:28 | 199.46.217.46 | trusted, issued by Entrust | 13110 |

domain `webvpn.raytheon.com`. We can observe the DNS resolution for the domain to switch to a secondary server for short periods of time, before switching back to the primary server.

- **Transition to new IP**

  After making a configuration change, the primary and secondary name servers reply differently until synced. We observe alternating DNS results for up to an hour, until both servers are synchronized.

- **Dynamic DNS**

  The DNS entry is updated with a new IP addresses whenever a new connection is established. This causes an update to the DNS entry every day at approximately the same time.

- **Configuration Changes and Errors**

  We observe a new IP address that is not reachable, and a consecutive change back to the previous address within minutes. One such example is shown in Table 5.2 for the domain `access.lockheedmartin.com`. The observed DNS changes are potentially caused by configuration changes or server maintenance.

**Table 5.2.** DNS resolutions and certificates for access.lockheedmartin.com

| Timestamp | IP addresses | Certificate | Cert ID |
|---|---|---|---|
| 2015-12-14 04:50:22 | 166.21.31.10 | | |
| 2016-02-04 06:40:03 | 143.114.76.200 | connection refused | |
| 2016-02-04 06:58:04 | 143.114.12.200 | trusted, issued by VeriSign | 21044 |
| 2016-02-11 22:42:53 | 143.114.76.200 | connection refused | |
| 2016-02-11 22:44:53 | 143.114.12.200 | trusted, issued by VeriSign | 21044 |
| 2016-03-10 12:43:08 | 143.114.76.200 | connection refused | |
| 2016-03-10 12:44:07 | 143.114.12.200 | trusted, issued by VeriSign | 21044 |

- **Anomalies**

  Some instances do not fit any of the previous categories. One example is shown in Figure 5.4, in which a domain is sometimes resolved to a web page containing ads instead of the companies webmail portal. The recorded DNS resolutions are shown in Table 5.3. Upon further investigation, we found that the domain `webmail.omni.com` uses a CNAME redirect to another zone. However, for some periods of time, the DNS server for the zone `omni.com.` will not only provide the CNAME, but also an A record for the CNAME, which is an IP address that points to an entirely different server.

- **Malicious**

  An attacker changes the A record to a new IP address under his control. The

**Table 5.3.** DNS resolutions and certificates for webmail.omni.com

| Timestamp | IP addresses | Certificate | Cert ID |
|---|---|---|---|
| 2016-01-13 14:26:28 | 207.74.79.184,207.74.79.185 | trusted | 30331 |
| 2016-01-27 12:15:43 | 141.8.225.31 | no TLS | |
| 2016-01-27 12:16:45 | 207.74.79.184,207.74.79.185 | trusted | 30331 |
| 2016-01-27 12:19:51 | 141.8.225.31 | no TLS | |
| 2016-01-27 12:21:52 | 207.74.79.184,207.74.79.185 | trusted | 30331 |
| 2016-01-27 12:24:55 | 141.8.225.31 | no TLS | |
| 2016-01-27 12:25:57 | 207.74.79.184,207.74.79.185 | trusted | 30331 |

IP address is reverted back to the original IP address after a short period of time. We could not find any certain examples of malicious changes.

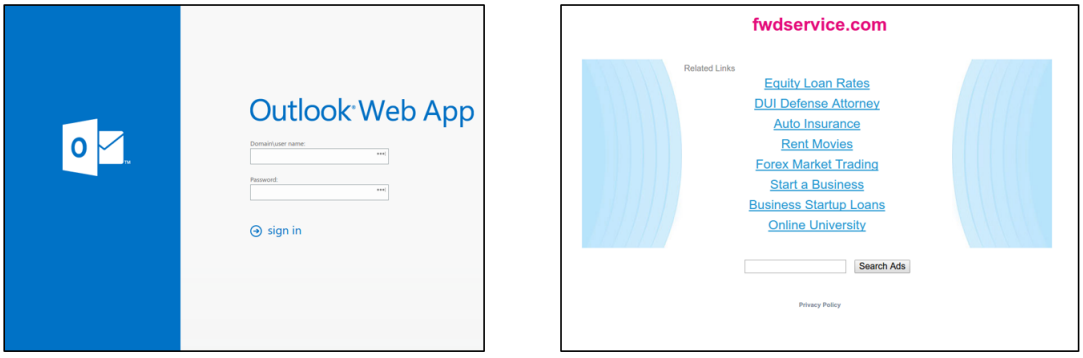


**Figure 5.4.** Example of an observed DNS anomaly

## 5.4 Comparing the resolved IPs with an IP blacklist

As a further analysis step, we took an aggregated IP blacklist collected from multiple data feeds of spam and malicious servers and looked for matches in the resolved IP addresses. Overall, 2581 IPs matched entries on this blacklist. Upon further analysis all of the matching IP addresses proved to be IP addresses associated with either cloud hosting providers or content delivery networks. It seems likely that, instead of these cases actually being malicious, a previous user of that host caused activity for it to appear on the blacklist. Given the questionable accuracy and high chance of false positives, the IP blacklist was of little use for finding compromised DNS entries.

## 5.5 Comparison with zone feeds

Another part of the analysis was the comparison of the results from the crawling with historic zone file data. Through the registrar Verisign, we were able

to access the zone files for both the .aero and .com zones to analyze historic NS and A glue records going back to the beginning of September 2014. The analysis of changes to NS and A glue records showed the following:

For the .aero zone, we found that 2042 out of 9620 NS servers changed during that time period. However, after classifying the web pages of the domains with changed NS records, we could show that 993 of these domains result in a failure page or are not accessible.

Similarly, for the .com zone, we found that 2570 out of 9620 NS servers changed during the time period, limiting the analyzed SLDs to the ones also found in the .aero zone. Similar to the .aero zone, we found that a large fraction of the domains with changes are parked domains (21%) or result in errors (31%). From this we assumed that a significant portion of these changes either happen when a company goes out of business, or when parked domains are transferred to another parking service. But even while ignoring the failure and parking cases, there is a large remaining number of domain names with changed NS records. Unfortunately, there are still a large number of other legitimate reasons, such as switching to a content delivery network, for such a change and there are no clear indicators to distinguish these from malicious changes.

## 5.6 Certificate data

During the analysis of the observed changes, some certificates proved to be of interest. A few domains were observed using multiple certificates. One pattern observed are regular certificate rotations. For example, Google's servers will use a new certificate each month. With the increasing popularity of Let's Encrypt, this could soon become even more common.

In one case, we can observe multiple certificates being used. Whenever the

primary server becomes unavailable, a backup system will take over. While the usual server is using a SHA256 certificate, the backup server still uses a SHA1 signed certificate which is going to expire at the end of the year.

Only few certificates for the same domain are issued by multiple certificate authorities. One frequent case are certificates issued by Cloudflare. The CDN has the ability to issue new certificates for customers through a Commodo CA. These certificates contain subject alternative names from multiple customers in the free version. On first look, these certificates raise suspicion.

## 5.7 Web crawling results

To validate that the crawled domains are related to the aerospace sector and are actually containing remote login pages, we analyzed the data collected through the website crawling. For this we used an already existing classifier from a previous project [30].

The classifier perform k-nearest neighbors classification on a simplified bag-of-words representation of the DOM content. While it is not perfectly accurate, it does provide qualitative results. The classifier created 279 clusters from the domains which were then labeled manually. An excerpt of the major clusters is shown in Figure 5.5. A total of 9,214 were part of clusters with a majority of login forms. This included generic login forms and webmail login interfaces. Another surprisingly large fraction of websites resulted either in HTTP errors or displayed a failure website. Some of these website were protected APIs or other protected resources. Additionally, we only tried to access the root path, which might have caused some of the HTTP 404 errors.

However, the largest clusters could not be clearly labeled, as they contained a diverse set of pages. From manual inspection, the majority of these seem to be

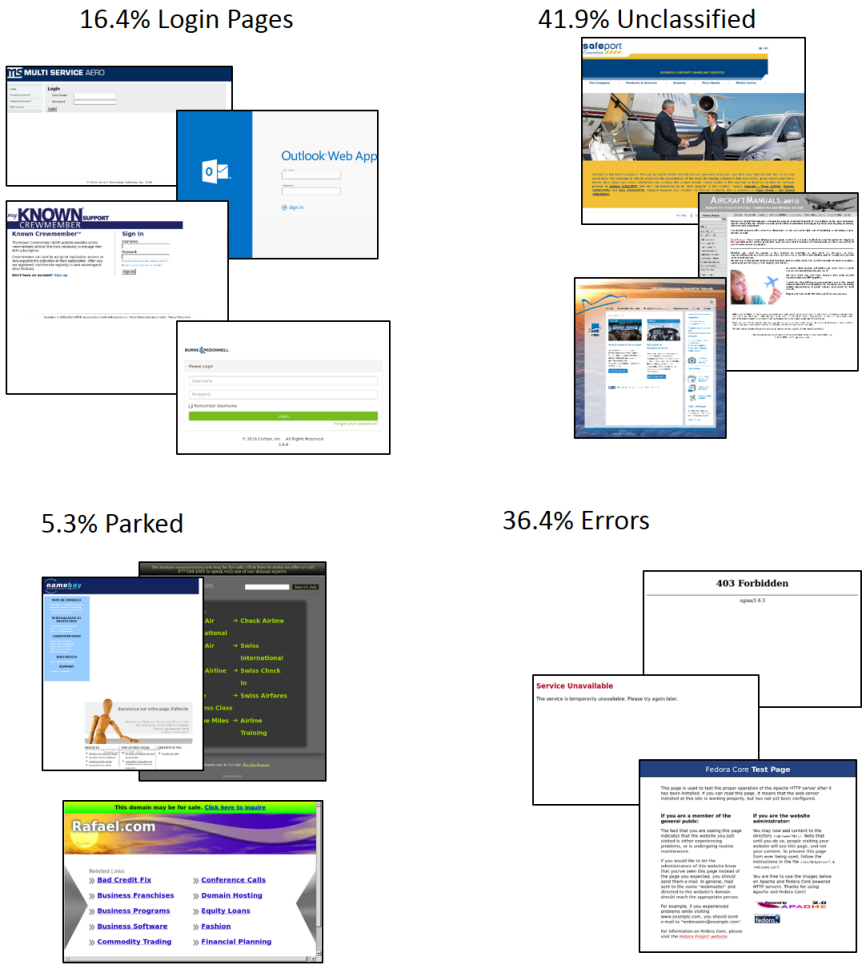


**Figure 5.5.** Domains clustered by website content using K-nearest neighbors algorithm.

generic landing pages for aerospace related companies. The detailed results of the clustering with a breakdown of the different categories is given in Table 5.4.

**Table 5.4.** Website categories

| *Category* | *# of domains* |
|---|---|
| Login page | 9,214 |
| Website with login form | 6,372 |
| IIS | 606 |
| Webmail | 2,236 |
| Outlook | 1,098 |
| SecureServer | 641 |
| Google | 295 |
| RoundCube | 202 |
| Parked Domain | 1,601 |
| NXDOMAIN redirect | 692 |
| Unconfigured web server | 704 |
| Unclassified | 23,572 |
| Errors | 20,466 |
| 400 Bad Request | 647 |
| 401 Unauthorized | 708 |
| 403 Forbidden | 2,939 |
| 404 Not Found | 1,742 |
| Unable to connect | 4,191 |
| Timeout | 8,082 |
| other errors | 2,157 |
| Total | 56249 |

## 5.8 Discussion

While some anomalies were detected, the results are somewhat dissatisfying, as proof of actual attacks is missing. This could mean that DNS hijacking attacks are very rare, or did not occur for any of the domains in our dataset. However, given that the DNS data is only collected over three months, and the certificate data for even less time, it seems likely that more interesting results will surface

during the course of further study. Nevertheless, we were able to detect interesting anomalies and were able to characterize most of the observed behavior.

# Chapter 6

# Conclusion

The security of the domain name system (DNS) is critical for protecting a targeted organization or enterprise. When a DNS registrar or provider is compromised, there are multiple ways an attacker can leverage this to gain access to user credentials, confidential internal emails, and other sensitive data. This thesis is a first attempt to evaluate and quantify how prevalent DNS hijacking attacks in the wild are. While we were able to gain some insights into common DNS behavior and configuration patterns through the active measurements, identifying actual attacks remains challenging. This chapter summarizes the results and insights gained from this work. Finally, we provide an outlook on potential limitations of this work and potential future work that one could consider pursuing.

## 6.1 Evolution of DNS hijacking

The given data seems to indicate that DNS hijacking through compromise of either the DNS registrar or the DNS hosting provider are becoming more and more frequent. Although most publicized attacks were relatively simple, usually involving website defacement or phishing attacks, recent reports and the details of the attack on Snecma seem to indicate that advanced persistent threats also utilize DNS hijacking in more subtle, less noticeable ways in order to attack high

value targets. However, these attacks are still rare compared to other typical attack vectors in breaches.

## 6.2 Active scanning results

The system developed within this thesis is able to detect even very short-lived changes to DNS entries. The additional components retrieving the certificate of a server and crawling the websites DOM content and screenshot are of further help for the forensic analysis of the data to distinguish between benign and malicious incidents.

As discussed previously, the initial assumption that most records in the DNS are static proved to be true. Over the measurement period of three months, 86% of the considered domains never changed. However, we also showed that patterns considered typical for the studied attack are far more widespread than initially assumed. For all categories of domains we investigated, short-lived DNS changes appeared to be the norm rather than an exception.

## 6.3 Limitations of a single vantage point

One limitation of this study is the limitation to a single vantage point. All measurements have been conducted from machines using the same address pool for all DNS queries. While the outgoing address pool of six /24 networks does provide some diversity of source address, these are still routed in the same way and likely to query the same regional DNS server. If an attacker indeed deploys their own DNS server, it would be possible to provide differing replies based on the source of the query. One possibility would be to only hijack requests originating from a certain geographical region using GeoIP technology. Therefore, the conclusions drawn from

these result might only apply to the specific region and network location they were measured from.

## 6.4 Challenges of identifying attacks

The analysis of the collected data shows that identifying attacks is more challenging than initially assumed. In the beginning of this study, our belief was that short-lived changes are a rare occurrence and that most records in the DNS are relatively static. While the results showed that for most domains, the results are indeed static, the first hypothesis proved to be wrong. For a number of reasons, short-lived changes are by far the most common ones we observed. The primary reason for this has been the use of DNS load balancing and in particular schemes that use round-robin DNS. While this is not quite in accordance with the design proposed in RFC 1794 [8], this kind of load balancing occurs frequently in our measurements.

However, even after filtering for load balancing and IP changes within the same subnetwork or changes to hosts with the same certificate, we still see other cases with attack-like change patterns. For some of these cases, we were able to identify possible explanations, including DNS failover in case of an outage, transitional periods between IP addresses, or configuration mistakes. In other cases, an explanation is not as obvious. However, we could not identify any clear signs for actual malicious attacks.

To summarize, given the evolution, adaptions, and optimizations made to DNS, the collected measurement results are diverse and complicated, making attack detection more challenging. More data over a longer period of time might help to identify additional patterns and additional details from actual attacks are required for a more accurate search for attacks.

## 6.5 Outlook

Given the aforementioned limitations of this work, the following areas could be of interest for further investigation

### 6.5.1 Increasing the scope of the study

As an initial pilot, the scope of queried domains was limited to the aerospace sector for better feasibility. However, as the data from Mandiant [34] and past attacks show, the aerospace sector is far from the only one being targeted by APTs and nation-states. Other potential sectors include U.S. and international governments, defence organizations and companies, health care providers.

A second improvement would be to extend the included domains from the specified sectors. A promising source for relevant domain names with likely external login pages are certificates published in the certificate transparency logs [28]. As the analysis by Jones [23] shows, there is only some overlap between the data collected by the Internet-wide TLS scans and the certificates in the transparency logs.

Both of these measures could help to increase the probability of observing ongoing attacks with the scans, but would come with additional challenges due to the increased amount of data.

### 6.5.2 Measuring from additional vantage points

As mentioned previously, currently all measurements are collected from a single university network. For the aforementioned reasons, it could be useful to have additional measurement locations. One potential way of achieving diverse, distributed measurements would be PlanetLab [12]. One could deploy the same measurements to multiple nodes in different networks and geographic regions to evaluate the importance of having multiple vantage points. However, to make this

feasible, one would have to make some optimizations to the queried set of domains first to reduce storage overhead and cope with DNS rate limiting.

### 6.5.3 Improving automated analysis

The analysis of the collected data that was presented in this thesis was based entirely on heuristics and manual labeling to distinguish benign changes from actual compromises of DNS registrars. This approach is not scalable, though, especially with an increased set of queried domains. A potential solution would be to use the IP block assignments published by the regional Internet registries as discussed by Antonakakis et. al. in [2].

Additionally, it might be possible to train machine learning classifiers on past behavior to detect anomalies. A further challenge for that is collecting enough reliable ground truth data to train a model.

# Appendix A

# Additional figures

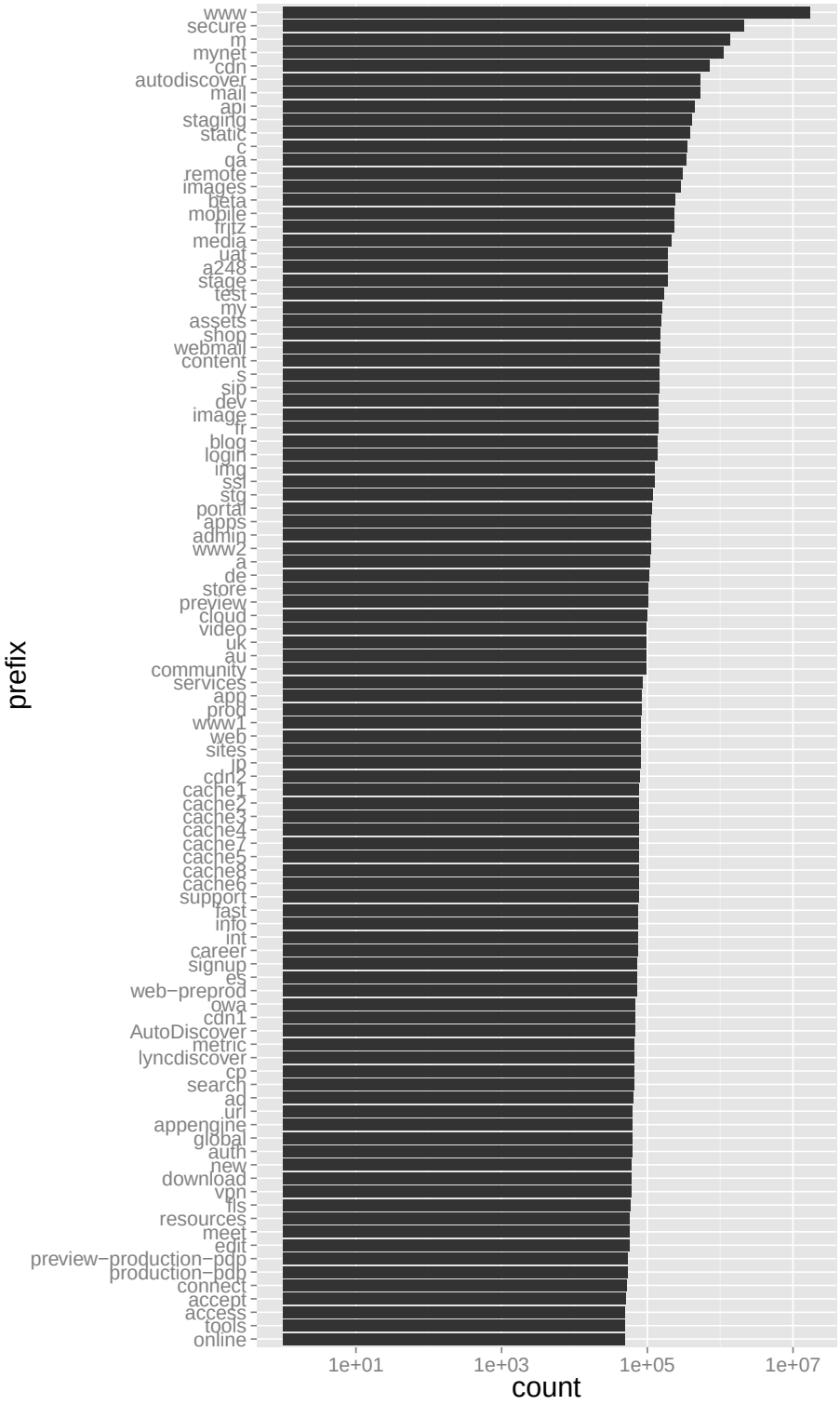


**Figure A.1.** Top 100 most frequent third-level domain components in TLS certificates

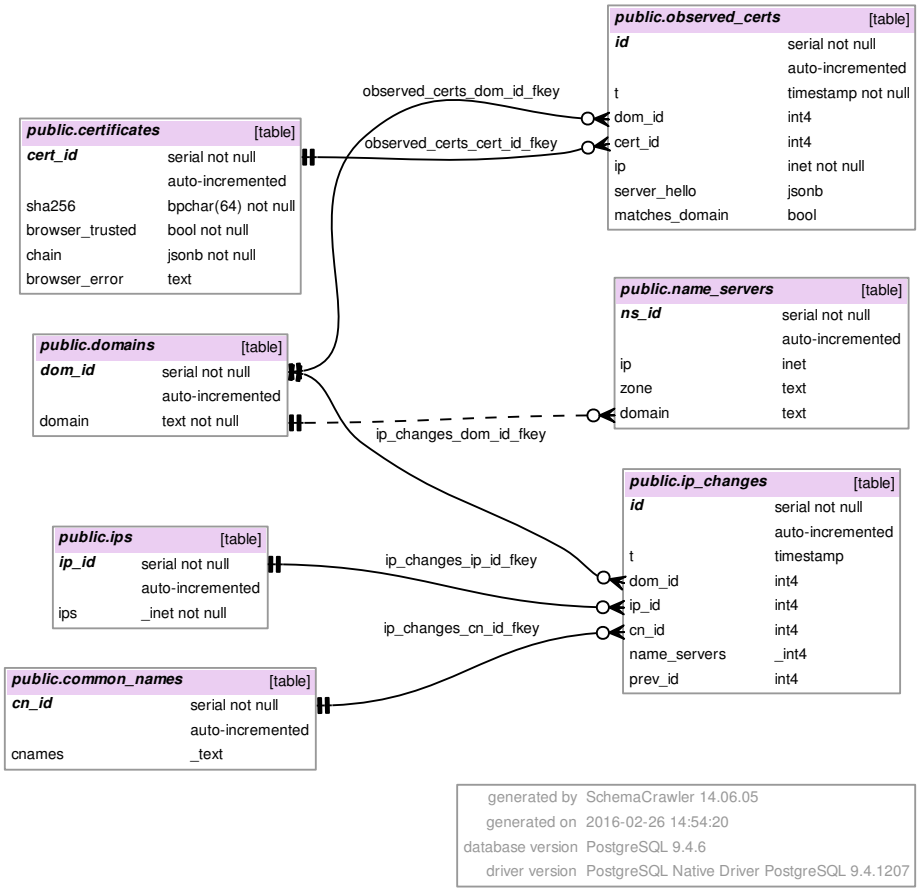


**Figure A.2.** Database schema for crawler results

# Bibliography


[1] CVE-2014-0322. https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-0322, February 14, 2014.

[2] Manos Antonakakis, David Dagon, Xiapu Luo, Roberto Perdisci, Wenke Lee, and Justin Bellmor. A centralized monitoring infrastructure for improving DNS security. In *Recent Advances in Intrusion Detection*, pages 18–37. Springer, 2010.

[3] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. DNS Security Introduction and Requirements. RFC 4033 (Proposed Standard), March 2005. Updated by RFCs 6014, 6840.

[4] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Protocol Modifications for the DNS Security Extensions. RFC 4035 (Proposed Standard), March 2005. Updated by RFCs 4470, 6014, 6840.

[5] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Resource Records for the DNS Security Extensions. RFC 4034 (Proposed Standard), March 2005. Updated by RFCs 4470, 6014, 6840, 6944.

[6] Simurgh Aryan, Homa Aryan, and J. Alex Halderman. Internet censorship in iran: A first look. In *3rd USENIX Workshop on Free and Open Communications on the Internet*. USENIX, 2013.

[7] S. Blake-Wilson, M. Nystrom, D. Hopwood, J. Mikkelsen, and T. Wright. Transport Layer Security (TLS) Extensions. RFC 3546 (Proposed Standard), June 2003. Obsoleted by RFC 4366.

[8] T. Brisco. DNS Support for Load Balancing. RFC 1794 (Informational), April 1995.

[9] Chris Brook. Craigslist back online following dns hijack. https://threatpost.com/craigslist-back-online-following-dns-hijack/109559, November 24, 2014.

[10] Lisa Brownlee. Report: Chinese hackers used opm data to steal us military intel; 'significant risk to us military'. http://www.forbes.com/sites/lisabrownlee/2015/09/19/report-chinese-hackers-used-opm-data-to-steal-us-military-intel-significant-risk-to-us-military, September 19, 2015.

[11] Ping Chen, Lieven Desmet, and Christophe Huygens. A study on advanced persistent threats. In *Communications and Multimedia Security*, pages 63–72. Springer, 2014.

[12] Brent Chun, David Culler, Timothy Roscoe, Andy Bavier, Larry Peterson, Mike Wawrzoniak, and Mic Bowman. Planetlab: an overlay testbed for broad-coverage services. *ACM SIGCOMM Computer Communication Review*, 33(3):3–12, 2003.

[13] Kyle Constable. Uconn website compromised, prompting users to download malicious program. http://dailycampus.com/stories/uconn-website-compromised-malicious-program, December 27, 2015.

[14] Lucian Constantin. Hosting provider leaseweb falls victim to dns hijacking. http://www.pcworld.com/article/2052680/hosting-provider-leaseweb-falls-victim-to-dns-hijacking.html, October 7, 2013.

[15] Matt Dahl. The french connection: French aerospace-focused cve-2014-0322 attack shares similarities with 2012 capstone turbine activity. http://www.crowdstrike.com/blog/french-connection-french-aerospace-focused-cve-2014-0322-attack-shares-similarities-2012, February 25, 2014.

[16] Adriene Dempsey. Password reset for st. louis fed research website user accounts. https://www.stlouisfed.org/news-releases/2015/05/18/password-reset-for-st-louis-fed-research-website-user-accounts, May 18, 2015.

[17] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard), August 2008. Updated by RFCs 5746, 5878, 6176, 7465, 7507, 7568, 7627, 7685.

[18] FBI Cyber Division. FBI cyber division bulletin on tools reportedly used by OPM hackers. http://www.databreaches.net/fbi-cyber-division-bulletin-on-tools-reportedly-used-by-opm-hackers, June 05, 2015.

[19] Zakir Durumeric, David Adrian, Ariana Mirian, Michael Bailey, and J. Alex Halderman. A search engine backed by internet-wide scanning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 542–553. ACM, 2015.

[20] Zakir Durumeric, James Kasten, Michael Bailey, and J Alex Halderman. Analysis of the HTTPS certificate ecosystem. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 291–304. ACM, 2013.

[21] Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. ZMap: Fast Internet-wide scanning and its security applications. In *Proceedings of the 22nd USENIX Conference on Security*, pages 605–620. USENIX, 2013.

[22] Bill Gertz. Report: Chinese hackers used opm data to steal us military intel; 'significant risk to us military'. http://freebeacon.com/national-security/opm-hack-part-of-large-scale-cyber-attack-on-personal-data/, July 16, 2015.

[23] James Jones. Early impacts of Let's Encrypt. https://tacticalsecret.com/early-impacts-of-letsencrypt/, 2016.

[24] Dan Kaminsky. Black ops 2008: It's the end of the cache as we know it. *Black Hat USA*, 2008.

[25] Darien Kindlund, Xiaobo Chen, Mike Scott, Dan Caselden, and Ned Moran. Operation snowman: Deputydog actor compromises US Veterans of Foreign Wars website. https://www.fireeye.com/blog/threat-research/2014/02/operation-snowman-deputydog-actor-compromises-us-veterans-of-foreign-wars-website.html, February 13, 2014.

[26] Jeremy Kirk. Lenovo, google websites hijacked by dns attacks. http://www.pcworld.com/article/2889392/like-google-in-vietnam-lenovo-tripped-up-by-a-dns-attack.html, February 26, 2015.

[27] Brian Krebs. St. Louis Federal Reserve suffers DNS breach. http://krebsonsecurity.com/2015/05/st-louis-federal-reserve-suffers-dns-breach, May 18, 2015.

[28] B. Laurie, A. Langley, and E. Kasper. Certificate Transparency. RFC 6962 (Experimental), June 2013.

[29] Timothy Lee. The new york times web site was taken down by dns hijacking. here's what that means. https://www.washingtonpost.com/news/the-switch/wp/2013/08/27/the-new-york-times-web-site-was-taken-down-by-dns-hijacking-heres-what-that-means, August 27, 2013.

[30] Kirill Levchenko, Andreas Pitsillidis, Neha Chachra, Brandon Enright, Márk Félegyházi, Chris Grier, Tristan Halvorson, Chris Kanich, Christian Kreibich, He Liu, Damon McCoy, Nicholas Weaver, Vern Paxson, Geoffrey M. Voelker, and Stefan Savage. Click trajectories: End-to-end analysis of the spam value chain. In *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, pages 431–446. IEEE Computer Society, 2011.

[31] John Leyden. Hacktivists dish out dns hijack to paypal, ebay. http://www.theregister.co.uk/2014/02/03/ebay_dns_hijack_sea, February 3, 2014.

[32] Wilson Lian, Eric Rescorla, Hovav Shacham, and Stefan Savage. Measuring the practical impact of dnssec deployment. In *Proceedings of the 22Nd USENIX Conference on Security*, pages 573–588. USENIX, 2013.

[33] Sean Lyngaas. Exclusive: The opm breach details you haven't seen. https://fcw.com/articles/2015/08/21/opm-breach-timeline.aspx, August 21, 2015.

[34] Mandiant. M-Trends 2015: A view from the front lines. https://www2.fireeye.com/rs/fireye/images/rpt-m-trends-2015.pdf, 2015.

[35] Joseph Menn. U.S. employee data breach tied to Chinese intelligence: sources. http://www.reuters.com/article/us-usa-data-breach-idUSKBN0OZ20Z20150620, June 19, 2015.

[36] Micheal Mimoso. Registrar in metasploit dns hijacking not duped by fax. https://threatpost.com/registrar-in-metasploit-dns-hijacking-not-duped-by-fax/102588, October 15, 2013.

[37] P.V. Mockapetris. Domain names - concepts and facilities. RFC 1034 (INTERNET STANDARD), November 1987. Updated by RFCs 1101, 1183, 1348, 1876, 1982, 2065, 2181, 2308, 2535, 4033, 4034, 4035, 4343, 4035, 4592, 5936.

[38] P.V. Mockapetris. Domain names - implementation and specification. RFC 1035 (INTERNET STANDARD), November 1987. Updated by RFCs 1101, 1183, 1348, 1876, 1982, 1995, 1996, 2065, 2136, 2181, 2137, 2308, 2535, 2673, 2845, 3425, 3658, 4033, 4034, 4035, 4343, 5936, 5966, 6604.

[39] Aviv Raff and Barak Gabai. 0-day attack targets aerospace engine manufacturer's remote users. http://www.seculert.com/blog/2014/02/0-day-attack-targets-aerospace-companys-remote-users.html, February 18, 2014.

[40] RS Ross. Managing information security risk. *Organization, mission, and information system view. Gaithersburg, MD*, 2011.

[41] Sushant Sinha, Michael Bailey, and Farnam Jahanian. Shades of grey: On the effectiveness of reputation-based "blacklists". In *Malicious and Unwanted Software, 2008. MALWARE 2008. 3rd International Conference on*, pages 57–64. IEEE, 2008.

[42] Sooel Son and Vitaly Shmatikov. The hitchhiker's guide to DNS cache poisoning. In *Security and Privacy in Communication Networks*, pages 466–483. Springer, 2010.

[43] Taylor Soper. Domain registrar eNom suffers DNS attack targeting 'large Internet infrastructure companies'. http://www.geekwire.com/2015/domain-registrar-enom-suffers-dns-attack-targeting-large-internet-infrastructure-companies, May 20, 2015.

[44] Olivier Thonnard, Leyla Bilge, Gavin O'Gorman, Seán Kiernan, and Martin Lee. Industrial espionage and targeted attacks: Understanding the characteristics of an escalating threat. In *Research in attacks, intrusions, and defenses*, pages 64–85. Springer, 2012.

[45] Paul Vixie. DNS complexity. *ACM Queue*, 5(3):24–29, May 2007.

[46] Paul Vixie. What DNS is not. *ACM Queue*, 7(10):43–47, November 2009.

[47] Paul Vixie and Vernon Schryver. DNS response rate limiting (DNS RRL). http://ss.vix.su/~vixie/isc-tn-2012-1.txt, April 2012.

[48] Duane Wessels. DNS cache poisoners lazy, stupid, or evil? In *36th Meeting of the North American Network Operators' Group (NANOG36)*, 2006.

[49] Xueyang Xu, Z. Morley Mao, and J. Alex Halderman. Internet censorship in china: Where does the filtering occur? In *Proceedings of the 12th International Conference on Passive and Active Measurement*, pages 133–142. Springer-Verlag, 2011.