2016 IEEE Symposium on Security and Privacy

Domain-Z: 28 Registrations Later

Measuring the Exploitation of Residual Trust in Domains

Chaz Lever[†], Robert Walls[*], Yacin Nadji[†], David Dagon[†], Patrick McDaniel[*], Manos Antonakakis[‡]

{chazlever,yacin,manos}@gatech.edu, dagon@sudo.sh, {rjwalls,mcdaniel}@cse.psu.edu

† Georgia Institute of Technology, School of Computer Science,
‡ Georgia Institute of Technology, School of Electrical and Computer Engineering,
∗ Pennsylvania State University, Department of Computer Science and Engineering

***Abstract*—Any individual that re-registers an expired domain implicitly inherits the *residual trust* associated with the domain's prior use. We find that adversaries can, and do, use malicious re-registration to exploit domain ownership changes—undermining the security of both users and systems. In fact, we find that many seemingly disparate security problems share a root cause in residual domain trust abuse. With this study we shed light on the seemingly unnoticed problem of residual domain trust by measuring the scope and growth of this abuse over the past six years. During this time, we identified 27,758 domains from public blacklists and 238,279 domains resolved by malware that expired and then were maliciously re-registered. To help address this problem, we propose a technical remedy and discuss several policy remedies. For the former, we develop Alembic, a lightweight algorithm that uses only passive observations from the Domain Name System (DNS) to flag potential domain ownership changes. We identify several instances of residual trust abuse using this algorithm, including an expired APT domain that could be used to revive existing infections.**

I. Introduction

Domain names have become the Internet's de facto root of trust. In practice, they are also a root of insecurity as common security systems depend on the unfounded assumption that domain ownership remains constant; this leaves users vulnerable to exploitation when domain ownership changes. For instance, authentication systems often rely on email to reset user passwords. Such schemes fail when the domain for that credential changes ownership—e.g., by expiration, auction, or transfer—and thus is no longer associated with the original owner. Consequently, an adversary can exploit this vulnerability to hijack the email address via a malicious re-registration of the domain.

In this paper, we study the exploitation of domain ownership changes and find that the phenomenon of *residual trust abuse* is the underlying cause of many, seemingly disparate, security issues. Among these, we found vulnerabilities allowing an attacker to maliciously register a domain to: *(i)* siphon University traffic and email by exploiting expired nameserver domains; *(ii)* hijack Regional Internet Registry (RIR) accounts and allocate IP addresses using expired email domains; and *(iii)* distribute malicious updates for benign software, including an instance that left users of a major Linux distribution vulnerable. The preceding examples demonstrate that even a single instance of residual trust abuse has major implications for the security of users and systems alike.

Despite the serious consequences of malicious registrations, the scope of the phenomenon has yet to be rigorously characterized and quantified. Our study seeks to fill this knowledge gap. Using data collected over six years, we show that *adversaries are actively exploiting residual trust*. To quantify this, we analyze the overlap between expired domains and both *(i)* hand-curated lists of malicious domains, i.e., public blacklists; and *(ii)* domains queried by malware, as such queries are an indicator of abuse. We find that almost 8.7% of the domain names that appeared on public blacklists (since 2009) were listed after the domains expired and changed ownership. In other words, over the last six years at least 27,758 were abusing residual trust. Similarly, we identified 238,279 domains that expired, were re-registered, and then contacted by malware—indicating likely malicious registrations. These domains account for 3.9% of all domains resolved by malware in our dataset. To put this into perspective, the size of this set is comparable to the 320,009 domains listed on public blacklists since 2009. Even more, empirical *evidence suggests this is a rapidly growing problem.* We found the exploitation of ownership changes has grown by orders of magnitude since we began collecting data. Between 2009 and 2012 there were 784 observed blacklist instances of abuse, but in 2014 alone, that number increased to over 9,000. We observed similar growth for expired domains resolved by malware, indicating this trend is not unique to blacklists.

In light of the increasing abuse of residual trust—e.g., malicious re-registration of domain names—better tools and policies are necessary to ensure the security of both users and systems. We argue that a comprehensive solution must consider both technical and non-technical remedies. For the former we propose Alembic, a lightweight algorithm that can be used to identify likely changes in ownership. This algorithm scales to large amounts of traffic, requires only access to historical DNS data, and ranks likely changes in domain ownership. Using our algorithm, we were able to identify several cases of potential residual trust abuse, including a currently expired advanced persistent threat (APT) domain. The expired APT domain example demonstrates how easily domains with negative residual trust can be used to revive existing infections. For the non-technical remedies, we discuss several potential policy changes

DOI 10.1109/SP.2016.47

and their implementation challenges.

Summarizing, our study makes the following contributions:

- We introduce the concept of residual trust and, using numerous real world cases of domain misuse, demonstrate how it is the underlying cause of many seemingly disparate security problems. Furthermore, we distinguish between positive and negative residual trust and discuss how each could be abused or cause unintended consequences.
- We provide the first large-scale analysis of residual trust abuse by using several large datasets for expired domains, passive DNS, network malware traces, and aggregated public blacklists. Our observations show malicious parties are actively abusing residual trust and that it is a growing problem.
- We propose a technical remedy and discuss several non-technical remedies to help deal with the growing abuse of residual trust. For the former, we introduce a lightweight algorithm, *Alembic*, to help locate likely ownership changes. Using our algorithm, we find several previously unidentified instances of abuse, including an expired APT domain.

While identifying changes in domain ownership would appear to be straightforward using WHOIS information [26], mining WHOIS is a challenging and resource-intensive task. Some researchers are trying to solve this problem with better automated solutions [36], but this does not address the problem that simply obtaining WHOIS information is expensive and hard to scale. Further, WHOIS information is rarely available in bulk. It is common for registry access to be limited to just a handful of queries (less than 1000) per day from a given host. While there are commercial companies offering limited API-based access to WHOIS information [16], [4], [15], they are cost-prohibitive and lack external validation. Due to the previously mentioned WHOIS limitations, it is outside the capabilities of most practitioners, research groups, and all but a handful of organizations to generate a comprehensive set of historic WHOIS records through which domain ownership changes can be identified.

These above constraints make building a traditional detection system for domain ownership changes extremely difficult. Therefore, we chose to create an efficient and highly scalable algorithm that helps find *potential* domain ownership changes using only DNS information.

II. Background

We define the term *residual trust* as the historical reputation of a domain that is implicitly transferred with changes in ownership. In this section, we detail the process governing a domain's expiration. In the following sections, we explain how these expired domains can be exploited by abusing the domain's residual trust.

Domain names are registered, owned, and expired using processes created by Internet Corporation for Assigned Names and Numbers (ICANN) in conjunction with registry operators and registrars. With a few exceptions, domains are typically registered for a period of one or more years, after which the registrant (i.e., owner) has the option to renew.

As a domain registration approaches its expiration date, it begins the formal ICANN expiration process. For generic top-level domains (such as .com, .net, and .info) the expiration process is governed by ICANN's Expired Registration Recovery Policy (ERRP) [33]. We summarize this process in Figure 1 and discuss the details below.

ICANN's expiration process is intended to address several past and potential abuses such as "domain sniping", whereby a vigilant "domainer" would *register* the domain seconds after expiration and extort a price to transfer the domain back to the former owner. Under the current process, domainers hoping to speculate on expired and lapsed domains must now wait until the release event, giving the current registrant time to renew the registration even after the domain expires.

Specifically, the ERRP requires registrars attempt to notify the lapsed owners (twice prior to expiration, once after). However, in practice, many owners cannot be reached due to a variety of reasons including inaccurate registration information, general neglect, or "tucked" domains. The latter reason, tucked domains, refers to situations where the contact information for the domain resides entirely under the expiring DNS zone itself. For instance, the registrar contact information, WHOIS information, and start of authority SOA RNAME [38] may be entirely under the expiring zone.

After the domain expires, the registrar will delete the domain from the TLD zone causing it to enter a 30-day Redemption Grace Period (RGP). Typically, deletion occurs within 1–45 days after expiration, but the exact length of time may vary due to extenuating circumstances or provisions in the myriad registrar and registry agreements. While in the grace period, the expired domain may still be renewed by the previous registrant, but this is typically at a higher cost. The domain is released five days following the conclusion of the RGP and becomes available for re-registration by others.

There are other variations of the domain expiration process. For example, the Canadian Internet Registration Association uses a "To Be Released" (TBR) process where expiring domains are listed along with all homonyms. For example, `cardreaders.ca` is TBR listed along with all accented variations such as `çardreaders.ca`, `cárdreaders.ca`, and other permutations. The 30-day process includes a short advance bid auction followed by general release.

Since many expiring domains are valuable brands, large groups of "drop-catchers" pool their resources to attempt registration in the first seconds after release. In order to prevent DDoS-style events against the registries, many providers stagger the release of expiring domains and publish the specific hour (and often the specific minute) during which a given domain will become available. Since valuable dropped domains are generally acquired within seconds, this strategy minimizes the period over which large volumes of registration attempts are directed against the registry.

Despite the post-expiration deletion phase, during which the domain is typically unreachable, third party users will often still attempt to connect to the domain. Increasingly, these connections are through automated tools, and users are often

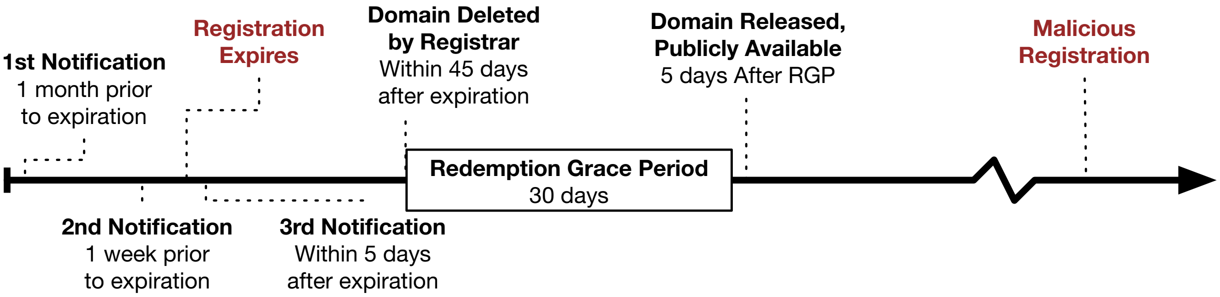

Fig. 1: **Timeline of a domain expiration.**

unaware the domain is even absent from DNS. For example, operating systems may attempt to update installed packages through an automated (e.g., cron, launchd) process. Browser plugins may contact home sites upon application startup. Software sharing tools may create connections to numerous file sharing sites on startup in order to obtain timely updates and routing tables stored in distributed hash tables. All of the domains associated with these automated activities can and do expire. Therefore, the party acquiring the expired domain has thousands and even millions of users contacting the site. We discuss specific examples and the security implications of this phenomenon in the next section.

III. Abusing Residual Trust

In this section, we discuss five real world examples of residual trust abuse that exploit expired domains previously used for a variety of Internet functions and services—including university DNS servers, CIDR allocations from Regional Internet Registries (RIRs), browser extensions, open source software, and promotional media content. These case studies demonstrate the unintended consequences that result from the residual trust placed upon domains by both users and systems. Our goal is to introduce the reader to the scope and severity of the problems caused by expired domains with concrete examples. Furthermore, these examples demonstrate that many seemingly disparate security issues actually share a common underlying cause: residual trust in domains.

A. Expired Nameserver Domains

In our first example, one of the DNS nameserver domains for the Benedictine University expired—potentially leaking sensitive university emails to the domain's new owners. According to our passive DNS sources, the `ben.edu` domain owned by Benedictine University used the following nameservers, among others, in 2012:

```
ben.edu. IN NS ns1.bobbroadband.com.
ben.edu. IN NS ns2.bobbroadband.com.
```

In other words, the hosts under `bobbroadband.com` provided secondary NS service for the university. It is common for organizations to rely on secondary DNS services from other organizations, often in different TLDs, to provide power and geographic diversity for their DNS. Consequently, the expiration of `bobbroadband.com` did not disrupt resolution of `ben.edu` as other DNS authorities were still available. Then, on October 25, 2012, the nameservers for `bobbroadband.com` were switched to the following:

```
bobbroadband.com. IN NS ns1.pendingrenewaldeletion.com.
bobbroadband.com. IN NS ns2.pendingrenewaldeletion.com.
```

The zone `pendingrenewaldeletion.com` is a special zone used by the registrar to manage the final stages of the domain through to the redemption grace period. The reader should note that the redemption grace period (described in Section II) is designed to cause an outage as a final way to notify a domain owner of an expiration. In this case, however, the redemption grace period process did not disrupt the university's DNS because other nameservers were still providing service. Ironically, the resiliency of DNS prevented the redemption grace period process from providing one last notice-through-outage to users.

After the domain expired completely, it was purchased by a search engine optimization (SEO) company that then responded to all domain queries with a wild-card answer. This directed all traffic destined for `ben.edu` (e.g., HTTP traffic, email, etc.) to an advertising site. These events are summarized in Figure 2.

This change is especially subtle because it was the domain of one of the nameservers for `ben.edu` that expired and not the university's own DNS record. Furthermore, the university still had other nameservers that would direct traffic to the school's servers, preventing the outage from occurring after every TTL for a given record. Thus, the outage intermittently manifested itself only if the nameserver handling a resolution was the one controlled by the SEO company—not one of the remaining authorities operated by the school.

Given the legal protections generally afforded to student emails, the ad company likely had no right to the traffic despite owning the domain. Clearly, there existed *residual trust* in the expired `bobbroadband.com` domain since an entire university depended upon it.

In a subsequent survey of the `edu` TLD, we identified nearly a hundred expired zones under the TLD. We offered our survey results of possible outages, similar to `ben.edu`, to the DNS community. An enterprise DNS company now provides secondary services for schools that formerly relied on expired

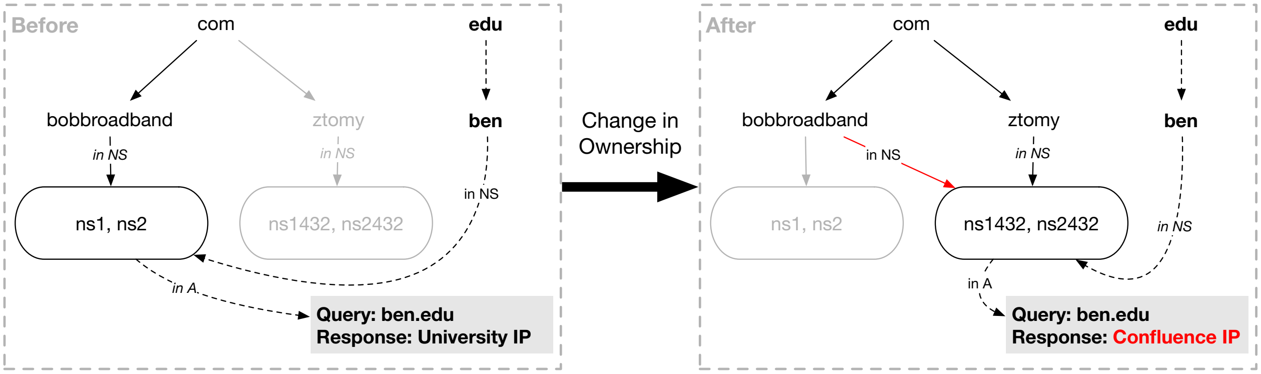

Fig. 2: Residual Trust Exploitation in University DNS Servers

or expiring secondary nameservers. While the problems caused in this example were many, the underlying cause was simple: *residual trust* in domains.

B. Expired Email Domains

In our second case study, we show how expired domain names could affect Regional Internet Registries (RIRs) operators. The RIRs locally administer the allocation of IP addresses [31] and maintain a database of which individuals have been allocated a specific Classless Inter-Domain Routing (CIDR) network. Stolen or hijacked RIR credentials can, therefore, lead to serious security incidents.

Account information for the RIR is protected using email as a trust anchor, and therefore, trust is effectively placed in whoever owns the domain specified by an email address. A simple check of the RIR databases yields all of the email addresses for CIDR operators, and registration checks on these domains indicated that hundreds of technical and administrative point-of-contact (PoC) listings were under expired domains.[1]

In all cases of expired contact details, we found either the `notify` or `abuse-mailbox` fields for `inetnum` and `aut-num` RIR objects contained emails under expired domains. One could simply register these domains, request a password reset, and log into the management interface to manage the allocated CIDRs. Indeed, there are several cases where this technique was abused to send spam [47].

We were in the process of notifying the various RIRs of our discovery when other researchers made public a technical report on this general problem [44]. Their work focused just on RIR objects, but we believe it supports our general focus on techniques to identify and manage expired domains. We continue to work through our RIR notification process and, therefore, omit listing the affected domains.

Like the previous case study, the underlying cause of this problem is residual trust. Email is regularly used as a trust anchor for online services and email addresses fundamentally rely on domains. Consequently, possession of a domain is often sufficient to demonstrate ownership of RIR CIDR allocations.

C. Expired Browser-Related Domains

Residual trust also offers an avenue for exploiting software. For example, many browser plugins contact one or more domains on startup to load both settings and content. To quantify this problem, we inspected approximately forty thousand plugins (many with different versions) from the Mozilla store. Specifically, we examined the online credentials of the authors, sites contacted by the plugins, and the author's contact information in the XPI manifest files. We found some 159 expired domains available for immediate registration.

Anyone could register one of these expired domains used by popular web browser plugins, some with tens of thousands of installations. This creates the possibility for a new owner to push updates to the plugin or to potentially take ownership of the associated developer account. While users may have trusted the original plugin developer, this trust should not extend to the new owners of the domains used by the plugin. This problem is exacerbated by the fact that users will be unaware of such ownership changes. Given that browser plugins can modify browser settings and behavior, this leads to potential security problems that are difficult to diagnose.

Our goal here is not to simply identify another browser plugin vulnerability. Other researchers have addressed other security aspects of browser extensions [21], [22], [35], [24] by analyzing the behavior and structure of browser plugins. Indeed, our analysis of this space was aided by the tools and frameworks noted above. Rather, this case identifies yet another instance of the unintended consequences caused by residual trust in domains. While existing work may stop potential abuse of this vector, we argue that the change in ownership of plugin domains is better dealt with by addressing the root cause: residual trust in domains.

D. Expired Open Source Software Domains

Residual trust from domain expirations also affects software repositories. Recently, the photo editing tool Gimp failed to renew its domain name, `gimp.org`. Fortunately, users noted the outage (days after the failed registration) [45] and

[1] To verify the expiration of each domain, we used a domain reseller account to access the parent registry via Extensible Provisioning Protocol (EPP) [32]. This step was necessary as DNS lookups resulting in `RCODE=3` or `NXDOMAIN` merely indicate the absence of records in a zone, not the availability of the record for registration. For a discussion of EPP use, we refer the reader to [30].

reported the problem. This allowed the domain to be recovered during the grace period—before a malicious registrant could obtain the domain and offer corrupted versions of the software.

A more disquieting outcome was seen in the recent "Debian multimedia" episode. For a while, an unaffiliated party operated an unofficial Debian repository mirror of multimedia applications (many of which did not meet the license requirements for the official Debian distribution). The domain `debian-multimedia.org` became popular and was linked to by various blogs, HOWTO articles, and software sites. Consequently, the site was added to the Advanced Packaging Tool mirror list for many Debian users. After some discussion with the maintainers of the official Debian distribution, the `debian-multimedia.org` owner agreed to create a new domain called `deb-multimedia.org` to avoid any indication of official endorsement. The previous `debian-multimedia.org` site later expired and was registered by a party unknown to the Debian community.

In effect, the new site owner had the ability to push software updates. This capability could be used to offer updates for even non-multimedia related packages such as the kernel or the base system. While a repository key system offered users the option to protect their updates, many users may choose to ignore warnings or may not have installed a key for the old site. This risk compelled the Debian maintainers to release a warning to end users instructing them to manually remove the old repository domain [48]. The notice alerted us to the problem, which we diagnosed as yet another symptom of a larger problem: *residual trust* in domains.

As noted above, there are protections against abuses in this dimension: software signing, local mirrors, staggered distributions in networks, rollbacks, and the like. But it is not clear if these solutions can be universally adopted by end users—many of whom simply wanted non-free multimedia software and followed well-intentioned but incomplete Internet resources. Instead of addressing the specifics of this challenging security area (the signing and verification of distributed software systems), we argue for a root-cause treatment of the problem: identifying changes in ownership of expired domains with *residual trust*.

E. Expired Spam Domains

In the previous cases studies, we examined cases where positive residual trust could be abused for malicious purposes, but we have yet to discuss the implications of domains carrying negative residual trust. Similar to benign domains, domains used for abuse often expire, and when this happens, they can be registered by new owners intending to use them for non-abusive purposes. But what happens when the new owner goes to share that newly purchased domain? Not surprisingly, the new owner may be censored by the same automatic safeguards put in place to protect online communities. Most maintainers of security lists or products will be completely unaware of ownership changes, and it may take a considerable amount of time before a domain is reclassified as non-abusive.

A public instance of this happened back in 2013 when Kirk Cameron released the film *Unstoppable*, a Christian movie targeting religious moviegoers [28]. A domain was purchased to market the film on the Internet, but this domain had previously been used to send spam—a fact presumably unknown the film's creators. Consequently, when this domain was used to market the film on Facebook, it was blocked by Facebook's automated spam detection systems. This led to heavily publicized outcries of censorship by the movie's producer and fans. Even after disclosing that the domain had been blocked by their automated spam detection systems, numerous articles decrying Facebook's censorship practices remained without update. Such claims of censorship, even after proven false, are a risk and a liability for a social network with millions of users of differing beliefs and world views.

Dataset Cardinalities

D_G	D_M	D_B	$D_M \cup D_B$
179,326,265	6,112,964	320,009	6,395,634

Datasets			**Dataset Intersections**		
A		B	$\%A$	$A \cap B$	$\%B$
D_G	$\cap$	D_B	0.1%	101,322	31.7%
D_G	$\cap$	D_M	0.2%	292,494	4.8%
D_M	$\cap$	D_B	0.1%	8,075	2.5%
D_G	$\cap$	$(D_M \cup D_B)$	0.2%	385,741	6.0%

TABLE I: In addition to the relative sizes of each set, this figure shows the relationships between the datasets of expired D_G, malware D_M, and public blacklist D_B domains.

Ultimately, this is yet another unintended consequence of the *residual trust* placed in domains. This incident could have been prevented if there were better systems in place to evaluate the trust associated with domains. Such systems could inform potential registrants of a domain's history before purchase or update security products after domain ownership changes.

IV. Measuring Residual Trust Abuse

In this section, we take a step back from looking at the specific cases of abuse and instead analyze the problem of residual trust abuse at scale. In particular, we analyze expired domains and malicious re-registrations from the past six years (2009–2015). We aggregate data from public blacklists, malware feeds, gTLD zone files, and other sources to measure the *scope* and *growth* of residual trust abuse. In summary:

- *Measuring Scope.* To measure scope, we identify and characterize expired domains associated with malicious behavior. In particular, we focus on expired domain names found on public blacklists or resolved by malware over the last six years. Our goal, in part, is to quantify the extent to which expired domains are exploited via malicious re-registration.
- *Measuring Growth.* For growth, we study the change in residual trust abuse over time by leveraging the temporal properties of our dataset. We measure when the domains expired and when they were used for abuse, allowing us to calculate the number of active instances of residual trust abuse.

Before diving into the results, we begin with a short discussion of the datasets used for our measurement study.

A. Measurement Datasets

Restricting our observation period to 2009–2015, we focus on the domains that were *(i)* observed to expire, *(ii)* placed on a public blacklist, or *(iii)* resolved by malware. The intersection between domains that expired and that were used for abuse yields sets of domains that are likely targets of residual trust abuse—possibly resulting in a malicious re-registration. In the following sections, we define these three sets of domains and provide greater detail about their contents.

1) Expired domains (D_G): We calculated the set of expired domains D_G by comparing successive gTLD zone transfers and recording removals. While the removal of a domain from a zone is a strong indicator of expiration, we further vetted such domains through the Extensible Provisioning Protocol (EPP) [32] using the domain reseller account noted in Section III. Finally, we augmented D_G with data obtained from a commercial drop-catch registration service [12].

Our D_G set consists of expired domains spanning November 2008 to July 2015 and contains 179,326,265 unique domains. Most commonly, the D_G domains expired due to the registrant's failure to re-register the domain. In a few cases, the domain changed ownership due to a trademark dispute [34], suspension, or registry action stemming from a court order.

2) Blacklist domains (D_B): The set D_B is an aggregation of eight public blacklists (Table II) collected from December 2009 to July 2015. As such, it includes several different types of malicious behavior from botnets to drive-by downloads. Importantly, D_B represents a *human-curated* list of domains associated with undesirable behavior. In total, there are 320,009 unique domains in this set. We use temporal information from our sources to determine whether a domain was added to a blacklist (D_B) before or after it expired (D_G).

3) Malware domains (D_M): D_M is a set of domains known to have been queried by malware. This set is compiled from three dynamic malware execution feeds: one academic and two commercial. These frameworks employ dynamic analysis to derive network and system indicators from binaries. These indicators often include URLs used for malicious purposes, e.g., command and control or advertisement fraud.

This dataset also contains temporal information for the malware execution (i.e., timestamp and DNS query), allowing us to determine whether the domain was used by malware before or after its expiration. D_M contains domains from seven years, occurring between the beginning of 2009 and July 2015, of malware execution traces from the aforementioned feeds and contains 6,112,964 unique domain names in total.

While not a guarantee of maliciousness, the domains logged by these systems adds a useful perspective to our analysis. This is especially true for those domains that appeared in a dynamic analysis trace *after* an ownership change. The reader should perceive this D_M set as an indicator, not a guarantee, of abusive behavior.

Blacklist	**Target**	**Source**
Abuse.ch	Malware, C&C.	[5]
Malware DL	Malware.	[13]
Blackhole DNS	Malware, Spyware.	[6]
sagadc	Malware, Fraud, SPAM.	[10]
hphosts	Malware, Fraud, Ad tracking.	[8]
SANS	Aggregate list.	[11]
itmate	Malicious Webpages.	[9]
driveby	Drive-by downloads.	[7]

TABLE II: Blacklist sources for D_B.

Expired Before Abuse

	D_Z	$D_G \cap D_M$	$D_G \cap D_B$
Num. of Domains	263,847	238,279	27,758
Avg. Days	888	911	692

Abused Before Expiration

	D_Z	$D_G \cap D_M$	$D_G \cap D_B$
Num. of Domains	123,396	54,215	73,564
Avg. Days	364	397	340

TABLE III: A breakdown of how many domains expired before and after abuse for expired blacklist ($D_G \cap D_B$), malware ($D_G \cap D_M$), and all abusive (D_Z) domains—as well as the average number of days between abuse and expiration.

4) Potentially abused expired domains (D_Z): Finally, we define the set of all domains that expired and were potential targets of residual trust abuse as $D_Z = D_G \cap (D_M \cup D_B)$.[2] In the context of this study, D_Z acts as an *upper bound* on the number of expired domains witnessed between 2009 and 2015 that appeared on human-curated blacklists or that were resolved by malware. A summary describing the relationships between each of the above datasets can be seen in Table I. In total, D_Z comprises 385,741 domains.

B. Measuring Active Residual Trust Abuse

In order to measure active instances of residual trust abuse, we focus on domains that have expired (D_G) and also appear on blacklists (D_B) or are resolved by malware (D_M). This set, D_Z, contains domains that are likely candidates for residual trust abuse through malicious re-registration of the domain. While the majority, 292,494 (75.8%), of the domains in D_Z were associated with malware resolutions, almost a third, 101,322 (31.7%), appeared on at least one hand-curated public blacklist. These numbers indicate that a substantial portion of the expired domains were manually linked with abusive behavior. This raises an interesting question. Did the expiration occur before or after abuse?

Table III summarizes the measurement observations behind the domain names that expired and also appeared in our public blacklist and malware datasets. From D_Z, we observed 123,396 domains that existed in $D_M \cup D_B$ *before* appearing in D_G. In short, these domains were used for abusive behavior before they expired. From this subset, 54,215 (43.9%) were contacted by malware and 73,564 (59.6%) appeared on public

[2] The Z in D_Z stands for zombie. Similarly, the G in D_G stands for graveyard. These identifiers, as well as the paper's title, are in reference to the similarities between reanimated (i.e., re-registered) domains and the depictions of zombies in popular media.

blacklists. Additionally, 4,748 (8.8%) of the domains contacted by malware also appeared on a public blacklist. Given their historical association with malicious behavior, these domains represent instances of *negative residual trust*.

Security practitioners can leverage domains with such trust for good by using them for different reconnaissance techniques like sinkholing. It is also important to note that negative residual trust can be used for malicious purposes as well. For example, an APT actor could use an expired spam-related domain to camouflage itself as a different type of threat; this would likely stymie discovery or attack attribution.

Conversely, we observed 263,847 domains that expired before appearing in $D_M \cup D_B$. More specifically, 238,279 (90.3%) domains were contacted by malware and 27,758 (10.5%) appeared on public blacklists only after expiring. Therefore, these domains represent cases of *positive residual trust* potentially being used for illicit activities. By registering expiring domains, bad actors can leverage the benefits of any positive reputation (such as brand and industry sector properties) previously held by a domain. Previously, we highlighted several concrete instances of this problem (Section III). This problem is worsened by the fact that benign domains often remain on whitelists after ownership changes due to the difficulty of discovering such events. This is highlighted by the fact that only 3,327 (1.4%) of the domains that expired before being contacted by malware ever appeared on a PBL.

To better understand the types of malware that might be abusing residual trust, we categorized some of the different types of malware observed in D_Z. Table V shows the top 10 malware types and families for the malware observed communicating with a simple random sample of 10,000 domains that expired and then were potentially used for abuse. Trojans are by far the most common type, with many generic types such as "malware" and "heuristic" following. The families are similarly dominated by heuristically determined labels and a few family specific labels. For example, VB.SMIS and Vobfus are generic labels for obfuscated malware written in Visual Basic. While there are instances where the MD5 is flagged as benign by the AV engines, most are malicious. As more evidence of maliciousness, 915 of the 1,559 registrars were used for registering privacy protected domain names to mask the registrant's email address and name. While there are legitimate reasons to use such a service, they are commonly employed by malicious actors to evade WHOIS attribution.

Finally, we provide a breakdown of the top-level domains (TLDs) in D_Z in Table IV. The distribution largely corresponds to the general popularity of each respective TLD. The potential exception is edu. We observed proportionally more edu domains being used for malicious purposes after expiration—possibly due to the inherent trust users place in the educational TLD.

C. Measuring Temporal Properties of Residual Trust Abuse

Next, we focus our analysis on the temporal properties of residual trust. We start by referring the reader to Figure 4, which shows the distribution of deltas between expiration and first indicator of potential abuse. On average, this delta was around a year for domains contacted by malware or appeared on blacklists. The extended length of this dormancy period suggests that it may take a considerable amount of time before the trustworthiness of the current domain owner can be ascertained. Therefore, not only must changes in ownership be detected but such changes should be monitored until the new owner's trustworthiness can be determined.

Expired to Malicious		*Malicious to Expired*	
TLD	**Count**	**TLD**	**Count**
com	214,019	com	85,409
net	27,621	net	15,954
org	9,648	info	9,287
info	5,575	org	5,869
us	2,671	biz	3,226
biz	2,185	us	2,458
ca	846	cn	989
cn	646	mobi	76
co	175	asia	56
edu	146	ca	45
mobi	80	edu	15
asia	35	co	11
de	20	de	1

TABLE IV: TLD frequency for domains in D_Z. This includes all domains that were used for abuse and expired at some point. In total, we observed 13 TLDs used by these domains.

Diving deeper into the domains that expired *before* being used for abuse, we find that the delta between the last indicator of abuse and the expiration event was roughly two years on average. The full distribution of these deltas can be seen in Figure 4 and shows two peaks, appearing approximately one year apart, for domains contacted by malware or appearing on public blacklists before expiring. The two peaks represent a small number of domains and are an artifact of shared expiration events for domains in $D_M \cap D_B$.

The long delay between last observed malware communication and expiration could be due to several factors. For example, in order to maximize the utility of malicious domains, malware authors may choose not to allow a domain to expire until the number of malicious connections to that domain drops below some threshold (i.e., the domain could still being monetized by the botmaster). Additionally, a malware author may choose to prevent a domain from expiring in order to restrict security practitioners from taking over the domain.

D. Measuring the Growth of Residual Trust Abuse

Figure 3 shows residual-trust abuse is becoming more common. The number of domains being contacted by malware after expiration grew from 6,138 between 2009 and 2012 to over 12,000 in just 2013. Similarly, the number of previously expired domains subsequently appearing on blacklists has grown from 784 between 2009 and 2012 to over 9,000 in 2014 alone. Further, more than 100 of these domains were ranked in the top 10,000 by Alexa on the day they were added to the blacklist. The horizontal striations in the figure are an artifact of malware collection and blacklisting processes. Namely, the feed operator may add many domains (possibly for the same threat) on the same day. Similarly, the vertical gap for December 2015 is the result of missing data stemming from technical issues with our collection framework.

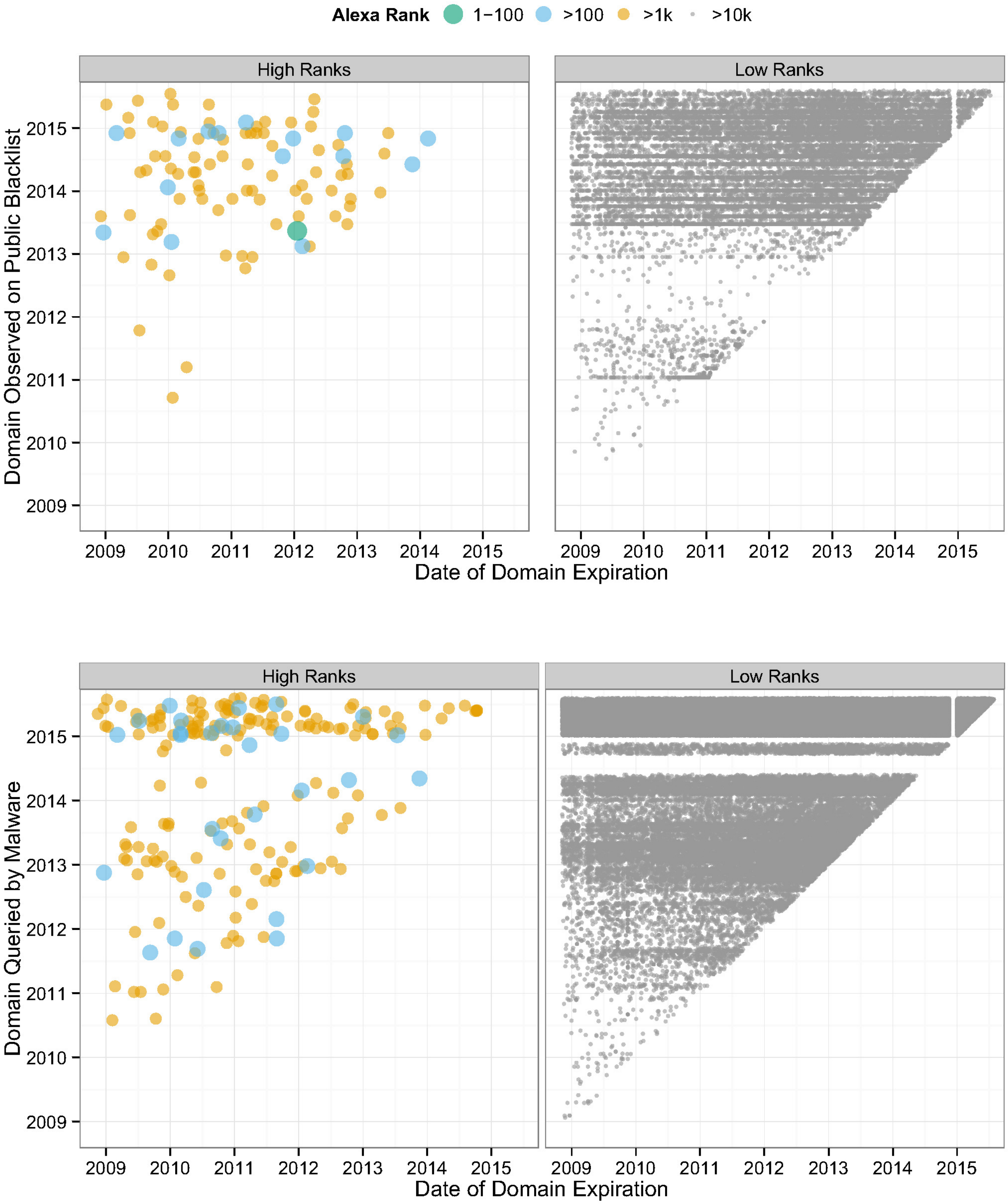

Fig. 3: Expiration date of a domain versus its first blacklist appearance or contact by malware. Each point represents one of the 27,758 (27.4% of D_B) or 238,279 (81.5% of D_M) distinct domains that expired and later appeared on a public blacklist; the dot's color corresponds to the domain's Alexa rank when it was added to the whitelist. The frequency of residual trust abuse has grown by multiple orders of magnitude since we began collecting data in 2009.

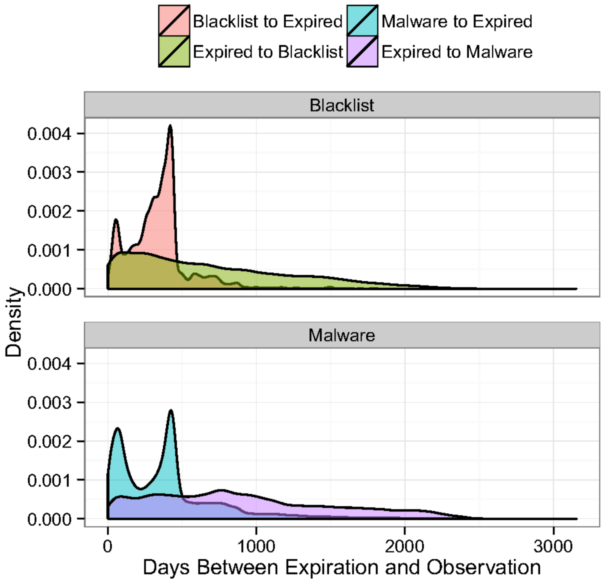

Fig. 4: Distribution of the number of days between domain expiration and contact by malware or appearance on a public blacklist. This figure shows there is often a significant dormancy period before the residual trust of a domain is abused.

Type	Count	Family	Count
Trojan	11,979	VB.SMIS	2,449
Malware	5,436	Heuristic	2,271
Heuristic	2,271	VBCheMan-A	2,001
Worm	783	Generic	1,573
Not Malicious	486	Vobfus.M	1,014
W32	286	StartP-HV	995
Backdoor	216	Paskod-A	868
Undesirable Software	191	Not Malicious	486
Virus	86	Paskod-D	481
Packed	43	Download Agent	413

TABLE V: Top 10 malware types and families from Kaspersky/Sophos/TrendMicro for 10,000 randomly selected samples from D_M.

V. Alembic

The previous discussion illustrated numerous studies, experiments, and anecdotes wherein expired domains with residual trust resulted in security problems or potential risks to users. While many of these cases might be remedied using well-founded research and existing technologies around phishing, DNS poisoning, and key management, it would be useful to have a system that *prevents* the problem from escalating in the first place. In this section, we leverage our previous findings to take the first steps toward such a system.

At first blush, the WHOIS protocol [15] appears to be an ideal candidate to address this question of identity. Unfortunately, WHOIS suffers from a number of limitations that make it ill-suited to deploy on a large-scale: (a) lack of verification of data, (b) expense in scaling queries across all registries and thick registrar WHOIS servers (many of whom limit queries to a handful per day); (c) lack of structure to data; and (d) lack of historical data in bulk form. We therefore explored techniques using other data more likely to be available to network operators: passive DNS logs.

The result of our efforts is *Alembic*, a general algorithm to assist in locating potential changes in domain name ownership and identifying reanimated domains. This algorithm scales without the need to mine resources such as WHOIS data and could be implemented by any network operators (or researchers) with access to DNS logs.

We measure the effectiveness of the *Alembic* algorithm using a multi-year passive DNS dataset obtained from commodity sources. As noted above, the goal of *Alembic* is to help identify potential changes in domain ownership without the expense and complexity of mining enormous volumes of WHOIS data.

Finally, we discuss two instances of residual trust abuse encountered during our evaluation. Interestingly, one of these included witnessing signs of an APT attack against sensitive networks. Finding trivially weaponizable APT domains was beyond our initial goals, but this result indicates the sensitivity of our approach and suggests other possible uses.

A. *Inputs to Alembic*

In order to scale and identify domains that have changed owners without expiring, we must devise an algorithm to assist in locating ownership change events, independent of WHOIS data. To achieve this, we rely on two datasets: a large passive DNS dataset to identify significant changes in infrastructure and client lookup volumes, and historic records of domain name *start of authority* to locate structural changes to the domain's zone.

There are many commercial and public passive DNS systems which collect and archive historic DNS traffic. Many organizations archive their own DNS answers, and security companies routinely maintain phone book type lists of where domains historically pointed. Readers not familiar with passive DNS may wish to consult [50]. Because we needed bulk quantities of passive DNS, we used a private collection instead of other publicly available DNS services (which often permit non-bulk API access).

Our dataset is sizable and includes historic resolutions that occurred across an entire North American ISP from January 1, 2011 to December 31, 2014. For each day, this data contains: all domain names that were resolved, the IP addresses they resolved to, and the total volume of lookups observed for a given DNS resource record (i.e., a domain name and IP address tuple). Using this data we can identify significant changes in domain name infrastructure and lookup volume—both attributes are useful for Alembic's operation.

Start of authority, or SOA, records specify authoritative information about a particular DNS zone. They contain the primary name server (the `MNAME` under RFC 1034 [38]), the email of the domain name administrator (or `RNAME` field), and

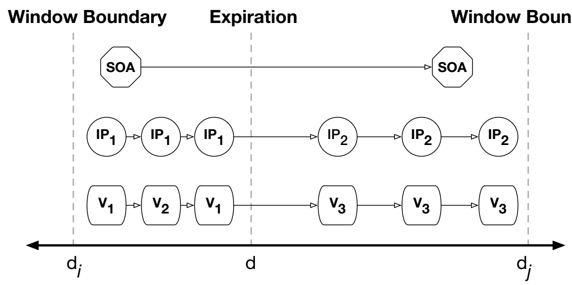

Fig. 5: Using different components to identify ownership changes.

other unique attributes of the zone.[3] Substantial changes in both the email and primary name server are strong indicators in ownership change for a given domain name. We therefore performed historical queries for SOA records for all the domains in D_Z.

B. Design of Alembic

We now describe how, using the aforementioned datasets, we identify domain names most likely to have undergone a change in ownership. We call our algorithm *Alembic*, after the still used by alchemists. *Alembic* lets us distill historical passive DNS evidence into a ranking of dates, and corresponding ranges, that are most likely to be associated with a change in domain ownership.

First, we discuss how we combine temporal changes in infrastructure, lookup volume, and SOA records into component scores. Then, we discuss how we generate the necessary inputs to compute these scores and how they are used to generate rankings of likely domain ownership changes.

1) Computing Component Scores: The *Alembic* algorithm is based upon the hypothesis that changes in ownership are highly likely to be accompanied by changes in network infrastructure, lookup volumes, and zone structure. While some users registering expired domains might be able to create the exact same zone content, host the nameservers at the same IPs, and generate the same SOA records, it is presumed this sort of subterfuge is both difficult and rare. This heuristic therefore comes down to the following conjecture: While one can perhaps buy any desired domain, one cannot so easily obtain its old IP address *and* use the same nameservers to manage the re-registered domain.

In order to identify these potential changes, the algorithm uses a temporal sliding window to measure changes in each component as observed in passive DNS resolutions over time. An overview of how the window and components fit together can be viewed in Figure 5. A summary of each individual component follows below.

Infrastructure Changes. For given a temporal window, W, we compute the Jaccard distance between hosts observed during the first and second portion of the window; this measures the dissimilarity between hosts seen during each period of time. In Algorithm 1, this measurement is computed by the INFRA-SCORE function. The computed score will range from zero to one where zero indicates the sets are exactly the same and one indicates that the two sets are completely disjoint.

Algorithm 1 Computing Component Scores

```
function INFRA-SCORE($h_i, h_j$)
    return 1 - JACCARD-INDEX($h_i, h_j$)
end function

function VOL-SCORE($v_i, v_j$)
    $t\_val, p\_val \leftarrow$ TTEST($v_i, v_j$)
    return $1 - p\_val$
end function

function SOA-SCORE($s_i, s_j$)
    $m_i, r_i \leftarrow s_i$
    $m_j, r_j \leftarrow s_j$
    $M \leftarrow \frac{1}{2}(1 -$ JACCARD$(m_i, m_j))$
    $R \leftarrow \frac{1}{2}(1 -$ JACCARD$(r_i, r_j))$
    return $M + R$
end function
```

Lookup Volume. Similarly, the distribution of lookup volumes for a given domain is split into two intervals for the current temporal window, W. We compute a t-test between the two distributions to measure if the null hypothesis (i.e., whether there is no relationship between them) is supported. This returns both a t-score and a p-value. The p-value ranges between zero and one with a lower p-value suggesting that the observed distributions are more likely to be consistent with the null hypothesis. Thus, a lower p-value suggests that the distributions are more likely to be different and a higher p-value suggests that the distributions are more likely to be similar. The VOL-SCORE function in Algorithm 1 shows that the volume score is computed as one minus the p-value which results in dissimilar distributions receiving a higher score.

SOA Differences. Like the previous two cases, we compute a score based on observations about the difference between the first and second portion of the current temporal window, W. In particular, we measure changes to SOA records observed during these two intervals. Each SOA record contains two fields of interest: an authoritative nameserver, `MNAME`, and an e-mail address, `RNAME`, for the individual responsible for the zone. We measure changes to each of these fields independently in order to finely measure changes in SOA records. Thus, we compute the Jaccard distance between the set of `MNAME`s observed in each portion of W, and separately, we compute the Jaccard distance between the set of `RNAME`s observed in each portion of W. The SOA-SCORE function, in Algorithm 1, shows how we compute the overall score for changes in SOA records, and like the previous component scores, higher values indicate there were more changes between the first and second portion of the temporal window.

2) Alembic Algorithm: The *Alembic* algorithm uses the component scores to generate rankings of likely domain own-

[3] Those not familiar with DNS zones and DNS record types may wish to consult [49].

Algorithm 2 Alembic Algorithm

```
function ALEMBIC($d, h, v, s$)
    $W \leftarrow$ window size

    if $|h| \geq W$ then
        $h_i \leftarrow \frac{W}{2}$ records before date $d$ in $h$
        $h_j \leftarrow \frac{W}{2}$ records after date $d$ in $h$
        $score_h \leftarrow$ INFRA-SCORE$(h_i, h_j)$

        $d_i \leftarrow$ minimum date for record in $h_i$
        $d_j \leftarrow$ maximum date for record in $h_j$

        $v_i \leftarrow$ lookup distribution between $[d_i, d]$ in $v$
        $v_j \leftarrow$ lookup distribution between $(d, d_j]$ in $v$
        $score_v \leftarrow$ VOL-SCORE$(v_i, v_j)$

        $s_i \leftarrow$ SOA records seen between $[d_i, d]$ in $s$
        $s_j \leftarrow$ SOA records seen between $(d, d_j]$ in $s$
        $score_s \leftarrow$ SOA-SCORE$(s_i, s_j)$

        return $score_h + score_v + score_s$
    else
        return 0
    end if
end function
```

ership changes. Algorithm 2 presents a pseudo-code implementation of the *Alembic* algorithm.

The first step in the algorithm is to choose a window W. This window defines the number of days worth of passive DNS data, around some date d, required for the algorithm to compute a change in ownership score. For example, if $W = 14$, then seven days worth of records before and after d are necessary for the algorithm to run; if insufficient records are available, the algorithm simply returns zero. In Algorithm 2, this process results in h_i and h_j, which are sets of hosts seen in A records $\frac{W}{2}$ days before and after d. These sets are used as the input to INFRA-SCORE to compute the infrastructure component score.

Since not all domains will have W contiguous days worth of records around d, the algorithm tries to pick the $\frac{W}{2}$ closest days before and after d. This may result in date ranges of varying size for each half of W. Therefore, we compute the date range for a window, W, by finding the minimum date, d_i, associated with the records in h_i and the maximum date, d_j, associated with the records in h_j.

We use the date ranges $[d_i, d]$ and $(d, d_j]$ to compute the lookup volume distributions for each portion of W around d. If we do not have lookup volumes associated with a date in one of these ranges, we assign it a lookup volume of zero; this imbues information about how frequently the given domain is resolved. The lookup volume distributions for each date range, v_i and v_j, are given as inputs to the VOL-SCORE to compute the lookup volume component score.

Next, the SOA records observed between the date ranges $[d_i, d]$ and $(d, d_j]$ are placed into two sets, s_i and s_j, and these sets are given as parameters to SOA-SCORE to compute the SOA component score.

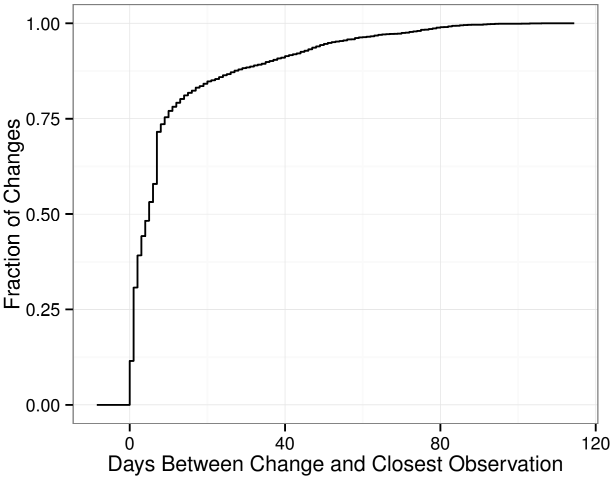

Fig. 6: CDF showing the distance (in days) between an ownership change and the closest observation in our passive DNS dataset. For 75% of the ownership changes, there is an observation in the passive DNS dataset that is less than 20 days away.

Finally, the change of ownership score is computed as the sum of each component score, which results in a value that ranges between zero and three. This score should be computed for each date that a passive DNS resolution was seen for a domain; these scores can then be sorted from highest to lowest to provide a ranking of dates, and corresponding ranges, which are most likely associated with changes in domain ownership.

The resulting list can be used to provide additional information about domains based on their residual trust. For example, whitelists can be pruned so that benign sites undergoing an ownership change can be quickly remapped to another appropriate category (e.g, “unknown” or “untrusted”) depending on the context. Knowledge of ownership changes can be leveraged to improve existing reputation and detection systems.

C. Efficacy of Alembic

Using the Alembic algorithm and our passive DNS dataset, we compute the ownership scores for a sample of *active* domains in D_Z. In our analysis, we define a domain as active if it was resolved at least W, with $W = 14$, times over any 120 day period in our dataset. This requirement filters domains for which the lack of observations would yield unreliable results. Similarly, we restrict our analysis to domains for which we were able to acquire ground truth about ownership changes. In total, we calculated 764,681 ownership scores for 11,564 domain names.

We compared the scores against known ownership changes gathered from archives of historically collected WHOIS data [16]: 17,838 changes in total. Figure 6 shows the distance between actual date of change and the closest observation date in our dataset. In short, 80% of the confirmed changes fall within 13 days of an observation in our dataset. This

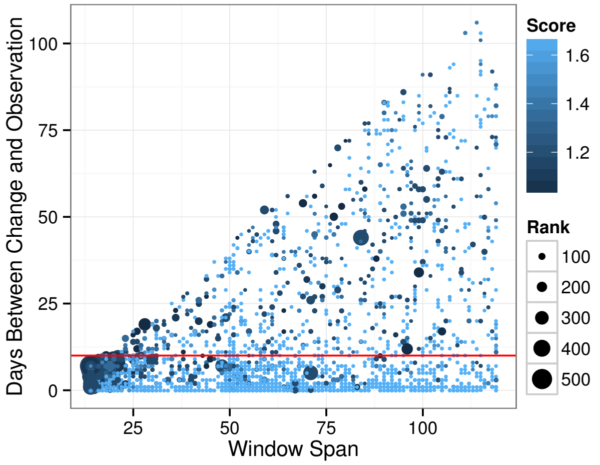

Fig. 7: Window timespan required for W observation days versus the distance between date of change and closest observation. This figure shows the best Alembic can perform given the sparse nature of the DNS resolutions for the domains in D_Z.

result is important as the effectiveness of Alembic depends on the frequency of DNS resolutions for a domain. Specifically, Alembic requires at least $W/2$ observation days before and after the candidate date. In other words, the span of the observation window depends on the resolution frequency of the domain. At a minimum, the window may span W consecutive days, i.e., the domain saw a DNS resolution on all W days. In the worst case, the domain may only be resolved once over our dataset's collection period. As mentioned above, we cap the date range necessary to collect W days of resolution behavior at 120 days. We show the date range of the observation window with respect to the number of days away from an exact match in Figure 7. In total, we find 4,543 (25.5%) of all changes fall within an Alembic observation window. Encouragingly, the bulk of these ownership change events occurred within ten days of an observation (red line in Figure 7)—even for larger observation ranges.

We believe our algorithm is a necessary step towards fostering additional research into domain ownership changes. Furthermore, our results show that Alembic, which works without relying on archiving and parsing WHOIS records, identifies potential changes in ownership. We plan to improve and refine Alembic to account for multiple ownership changes and sparsity in the input DNS data. For the latter, we propose investigation into the relationship between the frequency of resolutions for a domain and the span of the observation window required to detect ownership changes. Finally, we plan to explore other detection signals to use as component scores.

D. Additional Discoveries Using Alembic

We used Alembic to help identify abuses of both positive and negative residual trust. Here we discuss examples that fall into each of these categories. For the former, we highlight previously benign domains which were later used for command and control (C&C), leveraging the domain's historic reputation to exploit whitelisting. For negative residual trust, we highlight a potential attack vector whereby a leftover domain from a state-sponsored threat could be used to trivially gain access to sensitive networks where an infection has already occurred.

Date	Reg. Name	Reg. Email
10/15/08	Marcos Paulo dos Santos Fortunato	marcos.fortunato @contato.net
02/07/13	Identity Protection Service	doctorcompany.net @identity-protect.org

TABLE VI: Ownership changes to `doctorcompany.net`

1) Abuse of Positive Residual Trust: Here we study cases where Alembic helped identify cases of positive residual trust abuse. We present a brief look at two of the 263,847 domain names that were located by Alembic and subsequently became malicious only after expiring.

First we look at `doctorcompany.net`. After expiration, malware began using this domain for command and control (C&C). Anti-virus analysis from VirusTotal suggests this particular malware was variant of Win32/Polif [3] (a.k.a. Symmi). This particular threat is capable of numerous malicious activities including downloading and executing arbitrary files, logging keystrokes and other sensitive data, and exfiltrating any stolen information.

Using available historic WHOIS data, we estimate that `doctorcompany.net` changed owners once between 2008 and 2014. As shown in Table VI, the new owner chose to use an identity protection service when registering the domain, a common tactic used to by both legitimate and malicious users to exclude personal information from WHOIS records. Throughout the second lifetime of the domain and until its expiration—listed in the WHOIS record as February 7, 2014—the domain used the same nameservers, suggesting the owner remained the same during that year. We confirmed the domain became available for registration again on April 29, 2014—81 days after the listed expiry date and long enough to have passed through the entire expiration process described in Section II. About a month later on May 25, 2014, we saw malicious binaries attempting to query this domain. Since this domain had approximately six years of history without abuse, subsequent use by malware benefited from the domain's positive residual trust.

Similarly, `clicky.info` was also used for malware command and control (C&C) only after domain expiration. AV analysis suggests this particular malware sample is a variant of Win32/Nivdort [14], a trojan that steals key-presses, browsing history, credit card information and user-names and passwords. Using historic WHOIS, the domain's ownership appears to have changed eight times over the course of eight years. A summary appears in Table VII. Using historic WHOIS, we were able to confirm this domain was in pending delete status on February 11, 2014, and we subsequently confirmed its expiration on February 13, 2014 using the techniques mentioned in Section IV-A. As seen in Table VII, it was subsequently seen re-registered on March 9, 2014. The first observed communication by malware to this domain occurred

Date	Reg. Name	Reg. Email
03/16/06	Kim Fisher	jadothebest@hotmail.com
03/23/07	Derek Giordano	Derek@generalrate.com
01/01/09	Anders Oie	anders_oie@hotmail.com
04/05/10	Rubalier	cvx.conts@gmail.com
10/20/10	barry harding	bharding777@gmail.com
11/30/12	WANG SONGXU	sdwildcat@163.com
11/26/13	del del	del@del.del
03/09/14	Jeffrey Aikman	Roldvale@aol.com

TABLE VII: Ownership changes to clicky.info

on March 15, 2014—less than a week after being re-registered. Consequently, malware using this domain is able to leverage almost eight years of positive residual trust.

The WHOIS data for clicky.info shown in Table VII also highlights that that ownership changes are not always preceded by an expiration (domain registrations typically last at least one year). This further motivates the need for an algorithm like Alembic that helps locate ownership changes and illustrates the need for better awareness around the abuse of residual trust in domains.

2) Abuse of Negative Residual Trust: Next, we highlight a potential attack vector that leverages expired APT domains. On June 9, 2014 the security company CrowdStrike publicly released a report [2] detailing the cyber espionage activity of PLA Unit 61486. Also known as PUTTER PANDA, Unit 61486 is a branch of the Chinese SIGINT community.[4] Their mission, according to CrowdStrike, is to steal the trade secrets of corporations in the satellite, aerospace, and communication industries.

CrowdStrike's report identifies Chen Ping, as the primary persona responsible for obtaining domains for Unit 61486's C&C infrastructure. This moniker was derived from the registrant email stored in the WHOIS records, cpyy.chen-@gmail.com. We leveraged this knowledge to identify us-reports.net, an expired domain in our dataset that was previously registered using Chen Ping's email. We reanimated the domain, pointed it to a sinkhole, and found that despite being expired for years (and Unit 61486's activities being publicized in high-profile white-papers) our sinkhole began receiving connection attempts, every three seconds, from a national government research lab in Taiwan.

It follows that any malicious party with knowledge of the C&C protocol can capitalize on expired C&C domains to gain entry into already compromised networks—all for the low price of domain registration. This raises an important question: Should domains be available for re-registration after they were previously used for malicious purposes? We discuss this issue more in the following section.

VI. DISCUSSION OF POTENTIAL REMEDIES

Throughout this study, we have highlighted malicious re-registration and residual trust as the root cause of many seemingly disparate security problems. In Section III, we outlined several attacks and security lapses made possible by the abuse of this residual trust. Current solutions only address the symptoms of the underlying problem, not the cause, resulting in a plethora of techniques that only address narrow avenues of abuse. Instead, these problems would be better solved by addressing the underlying abuse vector.

In this section, we discuss potential remedies, both non-technical and technical, for residual trust abuse. Unfortunately, there is no single solution that can completely solve the problem; instead, a comprehensive remedy necessitates discussion and cooperation between all affected stakeholders. Our analysis of remedies is intended to outline the challenging nature of the problem with the hope it will foster further investigation by the security community.

A. Non-Technical Remedies

While any domain may carry residual trust, the severity of potential abuse is much greater for certain types of domains, e.g., those previously used by financial institutions or critical infrastructure. In short, domains that affect large numbers of users and systems, if abused, would benefit more from greater protections than other less important domains.

One potential remedy is to restrict critical industries to specially regulated zones. The idea is to limit who can register expired domains from one of these protected zones. Indeed, we already see this type of behavior with zones like gov and edu. Unfortunately, there are several unresolved questions and challenges with this solution. First, what criteria must be met for a domain to be considered critical? Second, how do we identify the existing critical domains? Third, assuming such domains could be identified, how do we migrate each domain from its existing zone? Finally, who is responsible for creating and managing the critical zones? These questions are made even more complicated by the global reach of the Internet; many diverse organizations (with different goals and motivations) would need to reach a consensus before any global solution could be adopted.

Rather than rely on custom zones, another potential option is to have the registrars or registries enforce special registration policies for critical domains. This solution is attractive as it could provide protection to critical domains under all zones and not simply those under a special top-level domain. However, this requires identification and reporting of all critical domains to either the registrars or registries and, for many organizations, this could be a challenging task. It also does not solve the problem of which domains qualify for protected registrations. This solution may be further complicated by the fact that any solution involving the registrars or registries also presumes that they would be willing participants. Given their financial interest in selling domains, there is a strong possibility that they would be reticent to employ any policies that make domain registration more cumbersome.

The previous two solutions focus on identifying critical domains; however, such solutions do not address the case where a non-critical domain is used as a trust anchor. For example, in Section III-B we saw how email addresses for expired domains were used for account management, thereby opening up the possibility for an attacker to hijack the account using malicious re-registration. For these domains, non-technical remedies need

[4] Unit 61486 is distinct from Unit 61398 described in Mandiant's APT1 report [1].

to be augmented with technical ones; we will discuss a couple such options in detail below.

B. Technical Remedies

When non-technical remedies fail, a technical solution is needed to mitigate problems. There are innumerable services that rely on third party domains, either for infrastructure or from users, and it is unlikely that many of these domains would fit some strict definition of a critical domain. As such, the non-technical policies proposed above would not be sufficient.

Instead, these systems should employ some process, such as Alembic (Section V), for identifying potential ownership changes. Such changes should be used to expire or revise the inherent residual trust of the associated domains. For instance, systems that rely on e-mail should re-evaluate access policies when e-mails expire or change ownership. A firewall rule that whitelists a domain should be revised to reclassify domains in order to avoid missing new attacks. A security information and event management (SIEM) device that classifies a domain as "low risk/spam/click-fraud/SEO" may revise the scoring of domains that have changed ownership. Given the active role of expired domains in APT attacks, this recommendation applies equally to forensic analysts and those investigating post-compromise events.

For smaller numbers of domains, it may be possible to use WHOIS to identify when the residual trust of domains should be re-evaluated, but this will not scale due to the complexities of bulk WHOIS collection. Furthermore, the lack of consistent formatting, use of privacy protection services, and inconsistent verification of WHOIS data may cause inferences relying on it to be unreliable. A system like Alembic could be used to address some of those concerns. In particular, it could be used to help identify ownership changes when scaling WHOIS becomes untenable, and since it relies on underlying network properties, it may find ownership changes that would be missed in WHOIS due to unreliable or forged data.

Dealing with residual trust is a challenging problem, but ignoring it exposes users and systems to a host of security issues. A comprehensive solution for this problem will require additional research and discussion by the security community.

VII. Related Work

There has been a wealth of research focused on using DNS as a tool for detecting malicious behavior. For example, researchers have previously used elements of DNS to classify malicious websites [37], [23]. Other researchers have used DNS information to understand and predict future malicious behavior [42], [43], [27], [29] and identify previously unknown malicious domains [19], [17], [18], [51], [40], [41]. In addition to using DNS for prediction and detection of malicious infrastructure, other work has focused on protecting the domain name system itself from abuse [25], [20]. Even commercial entities frequently use DNS-based tools to help protect against known malicious domains through the use of blacklists [46].

Our understanding of expired domain abuse first came from early research into the fate of failed banking domains by Moore and Clayton [39]. Their study focused on expired financial sites and found some instances where old, failed bank web sites were re-registered and likely used for nefarious purposes. However, the study authors were narrowly focused on methods for detecting failed banking domains.

Unlike this previous work, we study how *residual trust*—implicitly transferred between owners of a domain name—affects the security of systems and entities that rely on DNS. Our multi-year study demonstrates that residual trust abuse is being actively exploited and the problem is growing. Further, our work shows that this phenomenon impacts prior work by the security community and, thereby, demonstrates the need for more research into residual trust and malicious re-registrations.

VIII. Conclusions

Domains can change ownership for many reasons (e.g., expirations, auction, transfers) and the remaining *residual trust* is abused by clever attackers hoping to evade whitelists, hijack accounts, exploit software systems, or even buy access to existing infections. In short, we find that residual trust abuse is the root cause of many security issues on the Internet. At its core, there are potential policy and technical remedies. Policy remedies could identify potential avenues for exploiting residual domain trust and prevent or police re-registrations as appropriate. When that fails, technical remedies should actively try to identify ownership changes; we propose one such algorithm, Alembic.

Using a dataset of 179,326,265 expired domains spanning from December 2008 to July 2015, we quantify and characterize residual trust abuse and malicious re-registration. We found that 385,741 expired domains were contacted by malware or appeared on a public blacklist. This intersection contained almost a third, 101,322 (31.7%), of public blacklists domains in our dataset, and more troubling, a little over quarter, 27,758 (27.4%), of these domains expired before being blacklisted. In addition, only 3,327 (1.4%) domains contacted by malware after expiration ever appeared on a public blacklist. These findings demonstrate that the residual trust of expiring domains is being actively exploited. To make matters worse, we observe that the number of domains showing up on blacklists after expiration has grown from 784 between 2009 and 2012 to over 9,000 domains in 2014 alone; this shows that residual trust abuse is a growing phenomenon.

In order to help the research community flag potentially dangerous reanimated domain names, we developed a lightweight algorithm to rank potential domain ownership changes using only features that can be passively collected from DNS. We used this algorithm to identify several cases of residual trust abuse; specifically, we identified instances where re-registered domain names were used as infrastructure to facilitate attacks and one instance where an expired APT-related domain name could have been re-registered to gain access to an overseas government research lab.

IX. Acknowledgments

The authors would like to thank the anonymous reviewers for their insightful feedback and DomainTools for providing access to their Historic Whois Datasets. This work was supported in part by the US Department of Commerce under grant 2106DEK, and by the National Science Foundation under grants 2106DGX and CNS-1228700. Any opinions, findings,

and conclusions or recommendations expressed in this work are those of the authors and do not necessarily reflect the views of the sponsors.

References

[1] "APT1: Exposing One of China's Cyber Espionage Units," Mandiant, Tech. Rep., 2013, http://intelreport.mandiant.com/Mandiant_APT1_Report.pdf. [Online]. Available: http://intelreport.mandiant.com/Mandiant_APT1_Report.pdf

[2] "Putter Panda: PLA Army 3rd Department 12th Bureau Unit 61486," CrowdStrike, Inc., Tech. Rep., 2014, http://resources.crowdstrike.com/putterpanda/. [Online]. Available: http://resources.crowdstrike.com/putterpanda/

[3] "Backdoor:Win32/Polif.A," http://www.microsoft.com/security/portal/threat/encyclopedia/Entry.aspx?Name=Backdoor%3AWin32%2FPolif.A#tab=2, 2015.

[4] "Detailed domain name information and archives in one place," http://www.domainhistory.net/, 2015.

[5] "Domain Blacklist: abuse.ch," http://www.abuse.ch/, 2015.

[6] "Domain Blacklist: Blackhole DNS," http://www.malwaredomains.com/wordpress/?page_id=6, 2015.

[7] "Domain Blacklist: driveby," http://www.blade-defender.org/eval-lab/, 2015.

[8] "Domain Blacklist: hphosts," http://hosts-file.net/?s=Download, 2015.

[9] "Domain Blacklist: itmate," http://vurl.mysteryfcm.co.uk/, 2015.

[10] "Domain Blacklist: sagadc," http://dns-bh.sagadc.org/, 2015.

[11] "Domain Blacklist: SANS," https://isc.sans.edu/suspicious_domains.html, 2015.

[12] "Domain Graveyard," http://domaingraveyard.com/, 2015.

[13] "Malware Domain List," http://www.malwaredomainlist.com/forums/index.php?topic=3270.0, 2015.

[14] "TrojanDownloader:Win32/Nivdort.C," http://www.microsoft.com/security/portal/threat/encyclopedia/entry.aspx?Name=TrojanDownloader:Win32/Nivdort.C#tab=2, 2015.

[15] "Whoi.is," https://who.is/domain-history/, 2015.

[16] "Whois History," https://www.domaintools.com/research/whois-history/, 2015.

[17] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster, "Building a Dynamic Reputation System for DNS," in *Proceedings of the 19th USENIX Conference on Security (USENIX Security)*, August 2010.

[18] M. Antonakakis, R. Perdisci, W. Lee, N. Vasiloglou, and D. Dagon, "Detecting Malware Domains in the Upper DNS Hierarchy," in *Proceedings of the 20th USENIX Conference on Security (USENIX Security)*, August 2011.

[19] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou II, S. Abu-Nimeh, W. Lee, and D. Dagon, "From Throw-Away Traffic to Bots: Detecting the Rise of DGA-Based Malware," in *Proceedings of the 21st USENIX Conference on Security (USENIX Security)*, August 2012.

[20] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, "DNS security introduction and requirements," RFC 4033, March, Tech. Rep., 2005.

[21] S. Bandhakavi, S. T. King, P. Madhusudan, and M. Winslett, "VEX: Vetting Browser Extensions For Security Vulnerabilities," in *Proceedings of the 19th USENIX Conference on Security (USENIX Security)*, August 2010. [Online]. Available: https://www.usenix.org/event/sec10/tech/full_papers/Bandhakavi.pdf

[22] A. Barth, A. P. Felt, P. Saxena, and A. Boodman, "Protecting Browsers from Extension Vulnerabilities," in *Proceedings of the 17th Annual Network & Distributed System Security Symposium (NDSS)*, February 2010.

[23] D. Canali, M. Cova, G. Vigna, and C. Kruegel, "Prophiler: A Fast Filter for the Large-scale Detection of Malicious Web Pages," in *Proceedings of the 20th International Conference on World Wide Web (WWW)*, March 2011. [Online]. Available: http://doi.acm.org/10.1145/1963405.1963436

[24] N. Carlini, A. P. Felt, and D. Wagner, "An Evaluation of the Google Chrome Extension Security Architecture," in *Proceedings of the 21st USENIX Conference on Security (USENIX Security)*, August 2012. [Online]. Available: https://www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/carlini

[25] D. Dagon, M. Antonakakis, P. Vixie, T. Jinmei, and W. Lee, "Increased DNS Forgery Resistance Through 0x20-bit Encoding: Security via Leet Queries," in *Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS)*, October 2008. [Online]. Available: http://doi.acm.org/10.1145/1455770.1455798

[26] L. Daigle, "WHOIS Protocol Specification," RFC 3912 (Draft Standard), Internet Engineering Task Force, Sep. 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3912.txt

[27] M. Felegyhazi, C. Kreibich, and V. Paxson, "On the Potential of Proactive Domain Blacklisting," in *Proceedings of the 3rd USENIX Conference on Large-scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More (LEET)*, April 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1855686.1855692

[28] M. Gryboski, "Facebook Clarifies Reason for Blocking Kirk Cameron's "Unstoppable"," July 2013. [Online]. Available: http://www.christianpost.com/news/facebook-clarifies-reason-for-blocking-kirk-camerons-unstoppable-movie-site-100600/

[29] S. Hao, M. Thomas, V. Paxson, N. Feamster, C. Kreibich, C. Grier, and S. Hollenbeck, "Understanding the Domain Registration Behavior of Spammers," in *Proceedings of the 2013 Conference on Internet Measurement Conference (IMC)*, October 2013.

[30] S. Hollenbeck, "Extensible Provisioning Protocol (EPP)," RFC 5730 (INTERNET STANDARD), Internet Engineering Task Force, Aug. 2009. [Online]. Available: http://www.ietf.org/rfc/rfc5730.txt

[31] R. Housley, J. Curran, G. Huston, and D. Conrad, "The Internet Numbers Registry System," RFC 7020 (Informational), Internet Engineering Task Force, Aug. 2013. [Online]. Available: http://www.ietf.org/rfc/rfc7020.txt

[32] ICANN, "EPP Status Codes," https://www.icann.org/resources/pages/epp-status-codes-2014-06-16-en, 2015.

[33] ——, "Expired Registration Recovery Policy," https://www.icann.org/resources/pages/errp-2013-02-28-en, 2015.

[34] ——, "Uniform Domain-Name Dispute-Resolution Policy," https://www.icann.org/resources/pages/help/dndr/udrp-en, 2015.

[35] A. Kapravelos, C. Grier, N. Chachra, C. Kruegel, G. Vigna, and V. Paxson, "Hulk: Eliciting Malicious Behavior in Browser Extensions," in *Proceedings of the 23rd USENIX Conference on Security (USENIX Security)*, Aug. 2014. [Online]. Available: https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/kapravelos

[36] S. Liu, I. Foster, S. Savage, and G. M. Voelker, "Who is .com? Learning to Parse WHOIS Records," in *Proceedings of the 2015 Conference on Internet Measurement Conference (IMC)*, October 2015.

[37] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, June 2009.

[38] P. Mockapetris, "Domain names - concepts and facilities," RFC 1034 (INTERNET STANDARD), Internet Engineering Task Force, Nov. 1987, updated by RFCs 1101, 1183, 1348, 1876, 1982, 2065, 2181, 2308, 2535, 4033, 4034, 4035, 4343, 4035, 4592, 5936. [Online]. Available: http://www.ietf.org/rfc/rfc1034.txt

[39] T. Moore and R. Clayton, "The Ghosts of Banking Past: Empirical Analysis of Closed Bank Websites," in *Financial Cryptography and Data Security*, March 2014. [Online]. Available: http://ifca.ai/fc14/papers/fc14_submission_34.pdf

[40] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, "Phishnet: Predictive Blacklisting to Detect Phishing Attacks," in *Proceedings of the 29th Conference on Computer Communications (INFOCOM)*, March 2010.

[41] B. Rahbarinia, R. Perdisci, and M. Antonakakis, "Segugio: Efficient Behavior-Based Tracking of New Malware-Control Domains in Large Isp Networks," in *In Proceedings 45th Conference on Dependable Systems and Networks (DSN)*, June 2015.

[42] A. Ramachandran, N. Feamster, and D. Dagon, "Revealing Botnet Membership Using DNSBL Counter-intelligence," in *Proceedings of the 2nd Conference on Steps to Reducing Unwanted Traffic on the Internet (SRUTI)*, July 2006.

[43] K. Sato, K. Ishibashi, T. Toyono, and N. Miyake, “Extending Black Domain Name List by Using Co-occurrence Relation Between DNS Queries,” in *Proceedings of the 3rd USENIX Conference on Large-scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More (LEET)*, April 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1855686.1855694

[44] J. Schlamp, J. Gustafsson, M. Wählisch, T. C. Schmidt, and G. Carle, “The Abandoned Side of the Internet: Hijacking Internet Resources When Domain Names Expire,” Technische Universität München, Freie Universität Berlin, HAW Hamburg, Tech. Rep., December 2014. [Online]. Available: http://arxiv.org/abs/1412.5052v1

[45] M. Schumacher, “gimp.org domain has been renewed, DNS updates are still happening,” August 2015. [Online]. Available: https://mail.gnome.org/archives/gimp-developer-list/2015-August/msg00005.html

[46] Spamhaus, “DBL: The Domain Block List,” http://www.spamhaus.org/dbl/, 2015.

[47] ——, “SBL Advistory,” http://www.spamhaus.org/sbl/listings/RIPE, 2015. [Online]. Available: http://www.spamhaus.org/sbl/listings/RIPE

[48] D. P. Team, “Remove unofficial debian-multimedia.org repository from your sources,” https://bits.debian.org/2013/06/remove-debian-multimedia.html, 2013. [Online]. Available: https://bits.debian.org/2013/06/remove-debian-multimedia.html

[49] P. Vixie, “DNS Complexity,” *Queue*, vol. 5, no. 3, pp. 24–29, Apr. 2007. [Online]. Available: http://doi.acm.org/10.1145/1242489.1242499

[50] F. Weimer, “Passive DNS Replication,” in *In Proceedings of the 17th FIRST Conference on Computer Security Incident Handling*, June 2005.

[51] S. Yadav, A. K. K. Reddy, A. N. Reddy, and S. Ranjan, “Detecting Algorithmically Generated Malicious Domain Names,” in *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement (IMC)*, November 2010.