Dawn of the Dead Domain: Measuring the Exploitation of Residual Trust in Domains

Chaz Lever | Georgia Tech
Robert J. Walls | Worcester Polytechnic Institute
Yacin Nadji and David Dagon | Georgia Tech
Patrick McDaniel | Pennsylvania State University
Manos Antonakakis | Georgia Tech

An individual who re-registers an expired domain implicitly inherits the residual trust associated with the domain's prior use. Adversaries can, and increasingly do, exploit these ownership changes to undermine the security of both users and systems. As we enter the dawn of the dead domain, new techniques and policies are needed to fight this growing threat.

Domain names have become the Internet's de facto root of trust. In practice, they're 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—for instance, through 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.

These threats stem from the residual trust placed in domains, that is, a domain's reputation implicitly transferred with changes in ownership. For example, a domain previously used for benign purposes will often continue to be trusted even after it has expired or changed owners. Similarly, domains historically used to facilitate abuse will typically retain that negative reputation. This problem is engendered by the lack of effective mechanisms to identify when the domain's residual trust should be reevaluated. Without such indicators, users and systems that rely on domain names for security are subject to unacceptable risk.

Despite the theoretical possibility of residual trust abuse, how often are domains vulnerable to this threat? It turns out that domains on the Internet frequently expire or change ownership, creating many opportunities for abuse. In fact, there are even free services that help keep track of expiring domains over time; one such site estimated that approximately 22 million domains have expired in 2016 alone (domaingraveyard.com). In addition, numerous domain auction sites make it possible to acquire listed domains before they expire; a popular auction site reported that it facilitated 37,241 transactions through its marketplace in 2013—the most recent year for which public data is provided (sedo.com/us/buy-domains/market-trends). Although residual trust is a root cause of many problems, it's quite simple for an

 Copublished by the IEEE Computer and Reliability Societies

adversary to exploit in practice. All it takes is a simple re-registration of a previously expired domain.

We discuss several case studies that demonstrate how residual trust abuse is the underlying cause for several seemingly disparate problems. Beyond these examples, we measure the prevalence of this abuse through large-scale measurements using data collected over more than half a decade. This analysis shows that not only is residual trust abuse extremely common, it appears to be on the rise. To combat this growing threat, we discuss a new technical remedy that can help locate potential instances of residual trust abuse and augment this discussion with an examination of potential policy considerations.

DNS Basics

To understand residual trust abuse, it's necessary to first summarize the basics of DNS—a service that's widely used but often not well understood.[1] DNS facilitates most Internet communication. It serves as the phone book for the Internet, translating IP addresses into human-readable names and vice versa. Intuitively, DNS is based on the notion that most individuals can easily remember example.com but would struggle to memorize 192.168.15.25. DNS makes it possible for users to access desired resources using only easy-to-remember domain names. These domain names are organized as a hierarchical tree, where each level in the tree is separated by the "." character, and each part of the domain is referred to by its level. For example, the fully qualified domain name (FQDN) example.com has the top-level domain (TLD) of com and the second-level domain of example. The Internet Corporation for Assigned Names and Numbers (ICANN; www.dns.icann.org) is responsible for managing the list of valid TLDs.

Figure 1 shows the many steps involved in resolving a domain name into its corresponding IP address. In step 1, a stub resolver, located on the client, sends requests to a recursive DNS server, often simply called the *recursive*. If the answer isn't in the recursive's cache, the recursive iteratively queries different DNS name servers until it reaches the one containing the answer(s) for the current request, as seen in steps 2 to 7. This process always starts with a root name server, and at each step, the queried name server responds with the IP address of the name server responsible for the next level of the requested domain. This continues until the recursive reaches the authoritative name server, or simply *authoritative*, for example.com; the authoritative contains all the necessary information to translate the requested domain name into its corresponding IP(s), and it provides the DNS answer to the recursive. Finally, in step 8, the recursive forwards the response from the authoritative to the stub resolver and caches the response for a period of time dictated by the time-to-live.

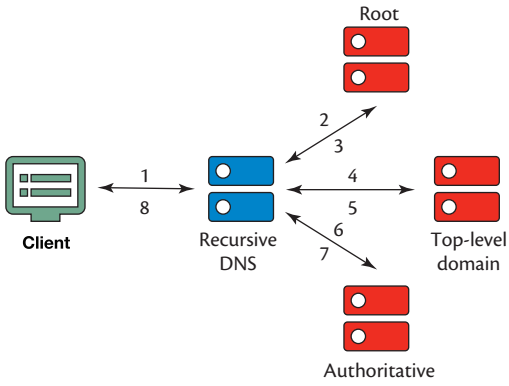

Figure 1. The steps to translate a domain name into its corresponding IP address. While the client sends a single query and receives a single response, there are many steps in the resolution process. Most of these steps are handled by the recursive DNS server.

Domain Expiration Process

Domain names are registered, owned, and expired using ICANN processes in conjunction with registry operators and registrars. With a few exceptions, domains are registered for a period of one or more years, after which the registrant (that is, the owner) has the option to renew.

As a domain registration approaches its expiration date, it begins the formal ICANN expiration process. For generic TLDs (such as .com, .net, and .info) the expiration process is governed by ICANN's Expired Registration Recovery Policy (ERRP).[2] Figure 2 summarizes this process.

ICANN's expiration process is intended to address several past and potential abuses, such as *domain sniping*, wherein a vigilant domainer registers a 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 to attempt to notify the lapsed owners (twice prior to expiration, once after). However, in practice, many domain owners can't be reached. This might be the result of inaccurate registration information, general neglect, or *tucked domains*, wherein the domain owner's contact information is under the expiring DNS zone itself. For instance, the registrar information for example.com might list the contact email as admin@example.com.

After the domain expires, the registrar deletes the domain from the TLD zone, causing it to enter a 30-day redemption grace period (RGP). Typically, deletion occurs between 1 and 45 days after expiration, but the

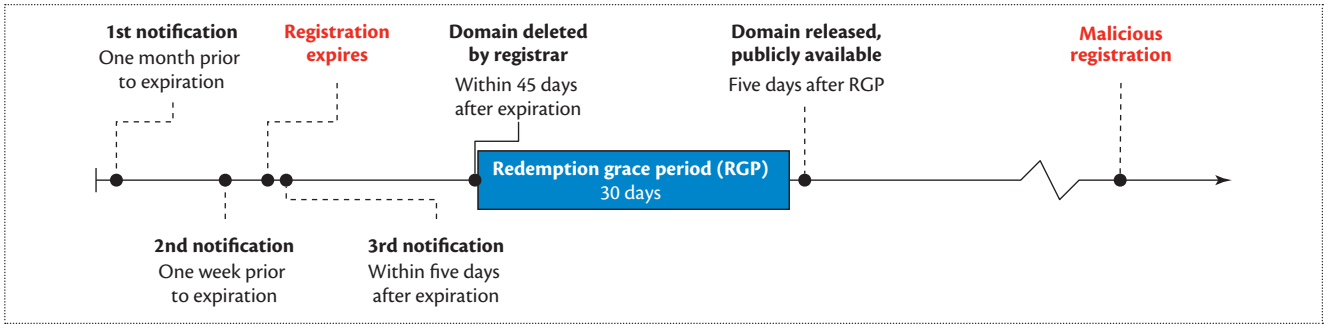

Figure 2. Timeline of a domain expiration. Notice that there are two grace periods: one after a domain has expired at the *registrar* and a second after the domain has been released to the *registry*. This affords a registrant many opportunities to reclaim a domain before it's released back to the public.

exact length of time can vary due to extenuating circumstances or provisions in the myriad registrar and registry agreements. While in the grace period, the expired domain can still be renewed by the previous registrant—typically at a higher cost. Five days following the conclusion of the RGP, the domain is released and becomes available for re-registration by others.

Often the expiring domains are valuable brands, prompting large groups of drop-catchers to pool their resources to attempt registration in the first seconds after release. To minimize the period over which large volumes of registration attempts are directed at the registry, 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.

Even after expiration, third-party users will often attempt to connect to the domain. Worryingly, these connections are increasingly made by background processes, and users are often unaware the domain is being contacted. For example, a piece of software might automatically contact the expired domain to check for updates. This behavior becomes a security concern when the expired domain is re-registered by a different owner. We highlight specific examples and the security implications of this phenomenon.

Residual Trust Abuse: A Source of Many Problems

Many unintended consequences result from changes in domain ownership. Although the issues often result in seemingly disparate security issues, they actually share a common underlying cause.

Expired Nameserver Domains

Organizations commonly rely on third-party DNS services, often in different TLDs, to back up and provide geographic diversity for their DNS. However, this practice backfired for Benedictine University when one of the domains for its third-party DNS nameservers expired but continued to be listed for the university's ben.edu domain. The expired nameserver domain was eventually purchased by a search engine optimization (SEO) company that proceeded to respond to all DNS queries with a wild-card answer. As a result, traffic destined for ben.edu—including HTTP requests and email—was redirected to an advertising site. These events are summarized in Figure 3.

This change was especially subtle because it was the domain of one of the nameservers for ben.edu that expired, not the university's own DNS record. Ironically, the resiliency of DNS prevented the RGP from providing one last notice-through-outage to users. Per the RGP, in October 2012, the nameservers for the expired domain were switched to a special zone designed to cause an outage: pendingrenewaldeletion .com. However, in this case, the change didn't disrupt the university's DNS because other nameservers were still available. Furthermore, the later outages caused by the SEO company's redirection only manifested if the nameserver handling a resolution was the one controlled by the company—not one of the remaining authorities operated by the school.

In a subsequent survey of the edu TLD, we identified nearly 100 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 or expiring secondary nameservers. Although the problems caused in this example were many, the underlying cause was simple: residual trust in domains.

Expired Email Domains

Expired domain names also affect regional Internet registry (RIR) operators. RIRs locally administer the allocation of IP addresses[4] and maintain a database of individuals that were allocated a specific classless

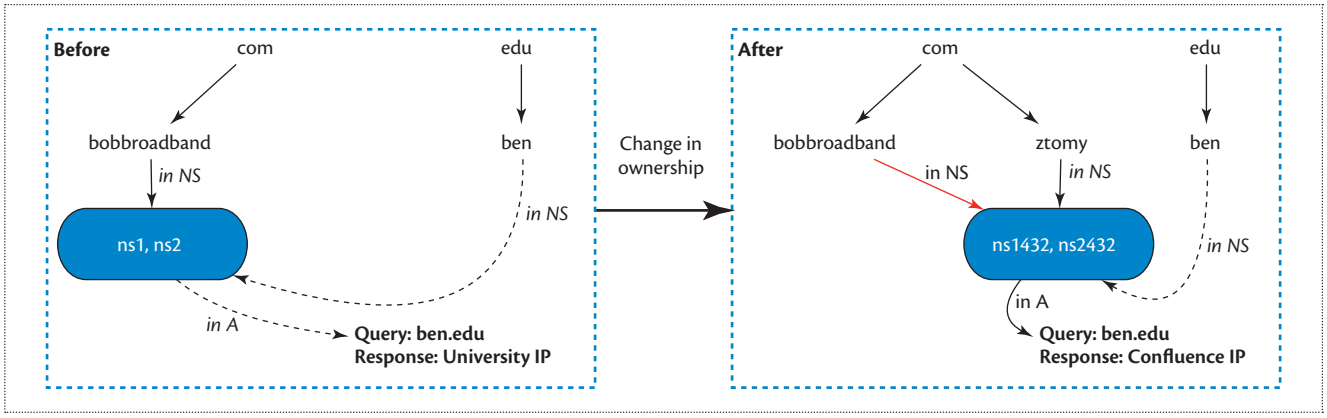

Figure 3. Residual trust exploitation in university DNS servers. Notice that the domain of the nameserver associated with ben.edu changed owners. After this change, ben.edu resolved to an IP address not controlled by the university, even though the university made no changes to its DNS records.

interdomain routing (CIDR) network. Account information for the RIR is protected using email as a trust anchor; therefore, trust is effectively placed in the owner of the domain specified by an email address. Stolen or hijacked RIR credentials can, therefore, lead to serious security incidents.

A simple check of the RIR databases revealed that hundreds of technical and administrative point-of-contact listings were under expired domains. One could simply register these domains, request a password reset, and log in to the interface to manage the allocated CIDRs. Indeed, in several cases this technique was abused to send spam (www.spamhaus.org/sbl/listings/RIPE).

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.

Expired Software Domains

Residual trust also offers an avenue for exploiting software. Recently, the photo-editing tool GIMP failed to renew its domain name, gimp.org. Fortunately, users noted the outage (days after the failed registration) and reported the problem.[4] 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 occurred with debian-multimedia.org. Debian is a popular Linux distribution known for its stability and focus on security. It comes with the Advanced Package Tool for managing the software installed on a computer and allows users to configure custom repositories for managing additional software packages. However, this particular site hosted an unofficial Debian repository of multimedia applications (many of which didn't meet the license requirements for the official Debian distribution). The domain grew in popularity and was linked to by various blogs, how-to articles, and software sites. After some discussion with the maintainers of the official Debian distribution, the owner of debian-multimedia.org agreed to create a new domain called deb-multimedia.org to avoid any indication of official endorsement. The previous domain eventually expired and was re-registered by a party unknown to the Debian community.

Because many Debian users added debian-multimedia.org to their Advanced Packaging Tool mirror list, the domain's new owner inherited the ability to push software updates. This capability extended to nonmultimedia-related packages, including the kernel and base system. Although the repository key system offered users some protection, the users could choose to ignore warnings or might not have installed a key for the old site. This security risk compelled the Debian maintainers to release a warning to end users instructing them to manually remove the old repository domain.[5]

Residual trust is also an issue for browser plug-ins. To measure the extent of the problem, we inspected approximately 40,000 plug-ins from the Mozilla store. Specifically, we examined the authors' online credentials and contact information and the sites contacted by the plug-ins. We found 159 expired domains used by browser plug-ins and available for immediate registration. Anyone could register one of these expired domains and push updates to the plug-in or potentially take ownership of the associated developer account. Worse, users would be unaware of such ownership changes. Given that browser plug-ins can modify

browser settings and behavior, this leads to security problems that are difficult to diagnose.

Abusing Negative Residual Trust

In the previous cases studies, we highlighted how a bad actor could exploit previously benign domains for malicious purposes. However, we have yet to discuss the implications of domains carrying negative residual trust—that is, what happens when a previously malicious domain is re-registered.

On one hand, the expired domain might be registered by new owners with benign intentions. Not surprisingly, the new owner might 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 might take a considerable amount of time before a domain is reclassified as nonabusive.

A public instance of this happened in 2013 when Kirk Cameron released the film *Unstoppable*, a Christian movie targeting religious moviegoers.[6] 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 to 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.

A new owner can also abuse the domain's negative residual trust for malicious purposes. On 9 June 2014, the security company CrowdStrike publicly released a report detailing the cyberespionage activity of People's Liberation Army (PLA) Unit 61486.[7] Also known as Putter Panda, Unit 61486 is a branch of the Chinese signal intelligence community (distinct from Unit 61398 described in the report). Its 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 command and control (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 usreports.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 to receive 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?

Measuring the Rise of Residual Trust Abuse

Beyond studying individual cases, we also measured the growth of residual trust abuse at scale using several historical datasets. Restricting our observation period to 2009 through 2015, we focus on the domains that were

- observed to expire,
- placed on a public blacklist, or
- resolved by malware.

The intersection between domains that expired and those used for abuse yields sets of domains that are likely targets for residual trust abuse—possibly the result of a malicious re-registration.

From these datasets, we estimate 179,326,265 domains expired between 2009 and 2015—again highlighting the many opportunities for abuse. Of those, we used historic blacklists and malware analysis feeds to associate 385,741 domains with malicious activity. This number indicates that a substantial portion of the expired domains were linked with abusive behavior and raises an interesting question: Did the expiration occur before or after abuse?

We observed 123,396 domains that were used for abusive behavior before they expired—that is, they were queried by malware or existed in a public blacklist only before expiration. From this subset, 54,215 (43.9 percent) were contacted by malware and 73,564 (59.6 percent) appeared on public blacklists. In addition, 4,748 (8.8 percent) of the domains were both contacted by malware and 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's important to note that negative residual trust can be used for malicious purposes as well. For example, an advanced persistent threat (APT) attacker 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 were used for abuse only after expiration. More specifically, 238,279 domains (90.3 percent) were contacted by malware and 27,758 (10.5 percent) 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. 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 percent) of the domains that expired before being contacted by malware ever appeared on a public blacklist.

These results suggest that residual trust is being abused, but they don't provide the temporal context to determine if such abuse is on the rise. Therefore, it's necessary to look at residual trust abuse at different points in time to determine if it's becoming more common. When we did this, we saw that the number of domains being contacted by malware after expiration grew from 6,138 between 2009 and 2012 to more than 12,000 in 2013. Similarly, the number of previously expired domains subsequently appearing on blacklists has grown from 784 between 2009 and 2012 to more than 9,000 in 2014 alone. To make matters worse, more than 100 of these domains were ranked in the top 10,000 by Alexa on the day they were added to the blacklist. Thus, not only do we see that residual trust abuse is increasing, but we see such abuse can even affect popular domains.

Finding Residual Trust Abuse

Although many of the cases we discussed could be remedied using established research and 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.

At first blush, the WHOIS protocol (who.is/domain-history) appears to be an ideal candidate to address this question of identity. Unfortunately, WHOIS suffers from several limitations that make it ill-suited to deploy on a large-scale:

- lack of verification of data,
- expense in scaling queries across all registries and thick registrar WHOIS servers (many of which limit queries to a handful per day),
- lack of data structure, and
- lack of bulk historical data.

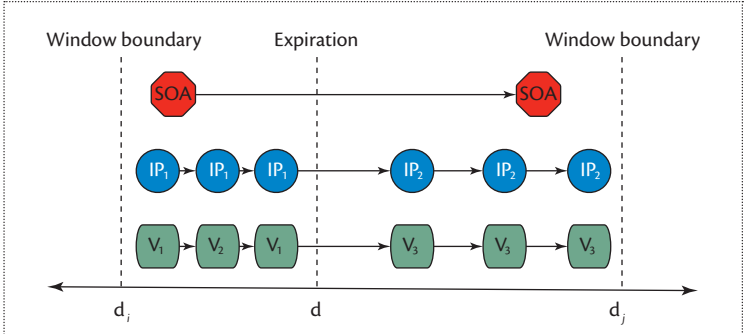

Figure 4. The sliding temporal window used by the Alembic algorithm, illustrating how individual components help locate potential ownership changes. SOA is Start of Authority records, V is volume, and d is date.

Therefore, we explored techniques using passive DNS logs—which are easier to acquire in bulk and more likely to be available to network operators.

The result of our efforts is Alembic, a general algorithm that helps locate potential changes in domain name ownership and identify reanimated domains. Named after the distillation apparatus 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. This algorithm not only scales to work with large numbers of domains but can also be implemented by any network operators (or researchers) with access to DNS logs.

The Alembic algorithm is based on the hypothesis that changes in ownership are highly likely to be accompanied by changes in network infrastructure, lookup volumes, and zone structure. Although some users registering expired domains might be able to host the nameservers at the exact same IPs, create the exact same zone content, and generate the same Start of Authority (SOA) records, this sort of subterfuge is presumably both difficult and rare. In short, although adversaries can perhaps buy any desired domain, they can't easily mirror its behavior.

To identify potential changes, the algorithm relies on three distinct components that describe a domain's infrastructure, lookup volume, and zone structure. Alembic uses a sliding temporal window to measure changes in each component as observed in passive DNS resolutions over time. Figure 4 is an overview of how the window and components fit together. Individual component scores are generated by measuring the changes between the two halves of the current temporal window, and the algorithm generates rankings of likely

domain ownership changes using an aggregate score—which is simply a combination of the individual component scores. Higher aggregate scores indicate a stronger likelihood of a change in ownership.

In our analysis, we found that the bulk of domains with high aggregate scores fell within 10 days of a verifiable ownership change—even for larger temporal windows. Thus, our algorithm helped us locate potential ownership changes with only passive DNS data; moreover, when likely changes were identified, it was also effective at providing a reasonable estimate of when that change occurred. We believe our algorithm is a necessary step toward fostering additional research into domain ownership changes.

Potential Remedies

We've highlighted malicious re-registration and residual trust as the root cause of many seemingly disparate security problems. Current solutions address the symptoms of the underlying problem, not the cause, resulting in a plethora of techniques that address only narrow avenues of abuse. Instead, these problems would be better solved by addressing the underlying abuse vector. Unfortunately, there's 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 and will hopefully foster further investigation by the security community.

Nontechnical Remedies

Although any domain might carry residual trust, the severity of potential abuse is much greater for certain types of domains, for instance, those previously used by financial institutions or critical infrastructure. Therefore, domains that affect large numbers of users and systems would benefit more from greater protections than other less important domains, and these protections could be addressed through new policies surrounding domain registration. Possible remedies include restricting critical industries to specially regulated zones or requiring registrars or registries to enforce special registration policies for critical domains. However, these solutions raise their own set of challenges including how to identify critical domains and who should be in charge of managing these domains. Even if solved, neither of these policy solutions address cases in which a noncritical domain is used as a trust anchor. For example, we discussed how email addresses for expired domains were used for account management—creating the possibility for an attacker to hijack the account using malicious re-registration. Nontechnical remedies need to be augmented with technical ones for these domains.

Technical Remedies

Technical solutions are needed to mitigate problems when nontechnical remedies fail. There are innumerable services that rely on third-party domains, either for infrastructure or from users, and it's unlikely that many of these domains would fit some strict definition of a critical domain. As a result, the nontechnical policies proposed earlier aren't sufficient. Instead, these systems should employ some process, such as our proposed Alembic algorithm, to identify potential ownership changes. Such changes should be used to expire or revise the associated domains' inherent residual trust.

For example, systems that rely on email should re-evaluate access policies when emails expire or change ownership. A firewall rule that whitelists a domain should be revised to reclassify domains to avoid missing new attacks. A security information and event management device that classifies a domain as low risk, spam, click fraud, or SEO should 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.

Dealing with residual trust is challenging, 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. ■

References

1. P. Mockapetris, "Domain Names—Implementation and Specification," RFC 1035 (Internet Standard), Internet Engineering Task Force, Nov. 1987; www.ietf.org/rfc/rfc1035.txt.
2. "Expired Registration Recovery Policy," ICANN, 2015; www.icann.org/resources/pages/errp-2013-02-28-en.
3. R. Housley et al., "The Internet Numbers Registry System," RFC 7020 (Informational), Internet Engineering Task Force, Aug. 2013; www.ietf.org/rfc/rfc7020.txt.
4. M. Schumacher, "Gimp.org Domain Has Been Renewed, DNS Updates Are Still Happening," Aug. 2015; mail.gnome.org/archives/gimp-developer-list/2015-August/msg00005.html.
5. D.P. Team, "Remove Unofficial debian-multimedia.org Repository from Your Sources," 14 June 2013; bits.debian.org/2013/06/remove-debian-multimedia.html.
6. M. Gryboski, "Facebook Clarifies Reason for Blocking Kirk Cameron's 'Unstoppable' Movie Site," *Christian Post*, 22 July 2013; www.christianpost.com/news/facebook-clarifies-reason-for-blocking-kirk-camerons-unstoppable-movie-site-100600.
7. *Putter Panda: PLA Army 3rd Department 12th Bureau*

Unit 61486, CrowdStrike tech. report, 2014; cdn0.vox-cdn.com/assets/4589853/crowdstrike-intelligence-report-putter-panda.original.pdf.

Chaz Lever is a PhD student in the College of Computing at Georgia Tech and a member of the Astrolavos Lab. His research interests include network and computer security. Lever received an MS in computer science from Wake Forest University. Contact him at chazlever@gatech.edu.

Robert J. Walls is an assistant professor in the Department of Computer Science at Worcester Polytechnic Institute. His research focuses on systems security and digital forensics. Walls received a PhD in computer science from the University of Massachusetts, Amherst. Contact him at rjwalls@wpi.edu.

Yacin Nadji is a computer security postdoctoral researcher with the Astrolavos Lab at Georgia Tech. His research interests include mobile device security, dynamic malware analysis, web security, and information retrieval. Nadji received a PhD in computer science from Georgia Tech. Contact him at yacin@gatech.edu.

David Dagon is a PhD student in computer science at Georgia Tech. His research interests include botnets, and he was involved in the development of ClickFox and is the inventor of its patented technology. Dagon received a JD from Florida State University College of Law. Contact him at dagon@sudo.sh.

Patrick McDaniel is a Distinguished Professor in the School of Electrical Engineering and Computer Science at Pennsylvania State University and the director of the Institute for Networking and Security Research. McDaniel received a PhD in electrical engineering and computer science from the University of Michigan. He's a Fellow of IEEE and ACM. Contact him at mcdaniel@cse.psu.edu.

Manos Antonakakis is an assistant professor in the School of Electrical and Computer Engineering at Georgia Tech. His research interests include computer and network security, anomaly detection, data mining, and attack attribution. Antonakakis received a PhD in computer science from Georgia Tech. Contact him at manos@gatech.edu.