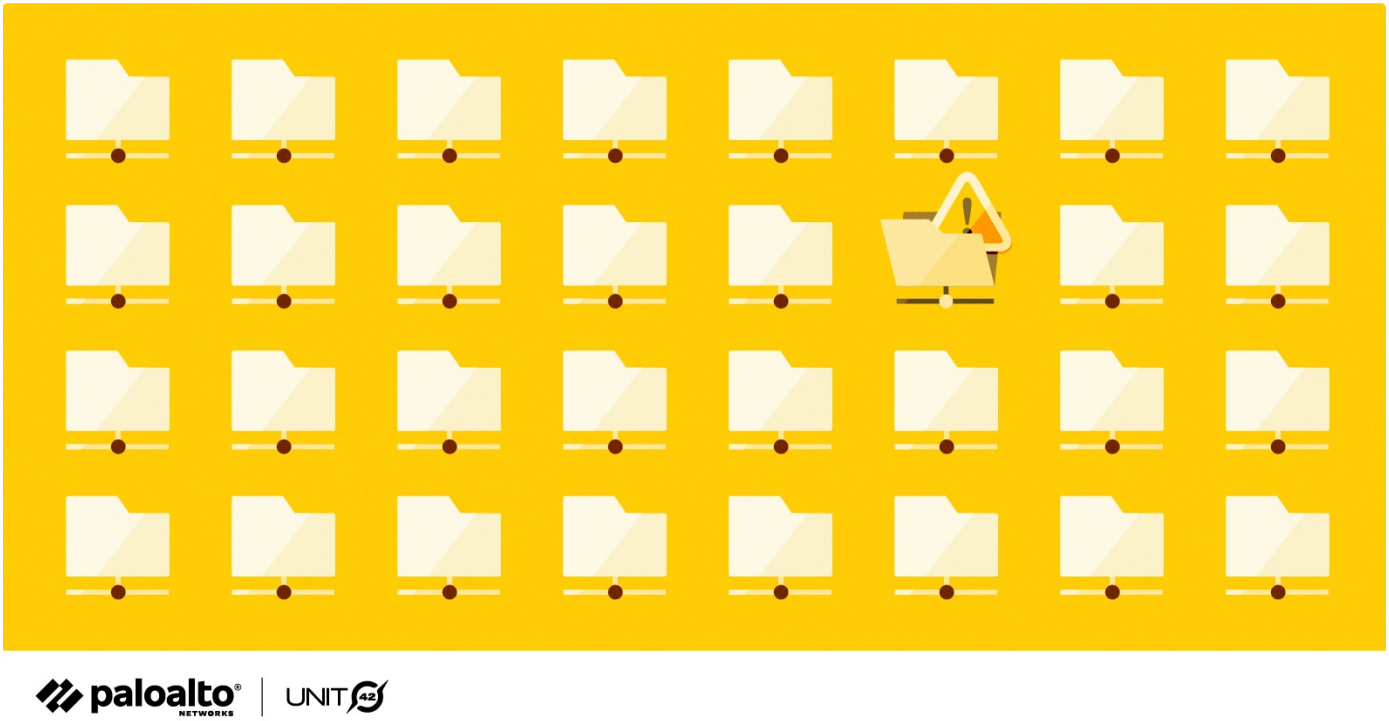
Dangling Domains: Security Threats, Detection and Prevalence

By Daiping Liu and Ruian Duan
September 16, 2021 at 6:00 AM
Category: Unit 42
Tags: dangling domains, DNS, DNS security, domain, domain hijacking, Phishing



This post is also available in: 日本語 (Japanese)

Executive Summary

The Domain Name System (DNS) provides the naming service which maps mnemonic domain names to various resources such as IP addresses, email servers and so on. As one of the most fundamental internet components, DNS and domain names usually serve as trusted anchors for users to access desired internet resources. As a result, threat actors constantly attempt to exploit DNS for illicit online activities. In particular, many attackers try to hijack domains with benign reputations. Several well-known techniques, including cache poisoning, malicious resolvers and domain registrar account hijacking, are used to achieve domain hijacking. However, great efforts like DNSSEC have been made to strengthen the DNS ecosystem in recent decades, and these hijacking techniques have become more challenging to achieve in practice.

Instead, a recent study has shown that a largely overlooked threat in DNS – dangling DNS records – could be easily exploited for domain

hijacking. In this blog, we will introduce several types of dangling DNS records and multiple techniques that can be used to exploit the dangling records. We built a detector that can actively identify dangling records from our collected DNS data.

Our results show that the dangling domain is a real and prevalent threat. Specifically, we have detected 317,000 unsafe dangling domains in total in our passive DNS data set. More worrisomely, even well-managed DNS zones have a non-trivial number of dangling domains. In particular, we found 13 dangling domains under the top-level domain (TLD) gov, 197 under the TLD edu, and 4,767 are detected under the Tranco top 2,000. All of these high-profile domains could be exploited for attacks like phishing and scams.

To protect users, once our detector identifies a dangling domain, the knowledge is distributed to multiple Palo Alto Networks security subscriptions, including DNS Security and Advanced URL Filtering. We also actively notify the corresponding domain registrars of the affected domains and the affected domain owners if the whois data is available.

Dangling Domains: An Overlooked Security Threat

A DNS record is essentially a pointer, where the rrname points to the network resource represented in rdata. When a resource in rdata is abandoned and released, the DNS record becomes dangling, and the rrname is called a dangling domain. If the abandoned resource could potentially be controlled by anyone besides the owners of the rrname, this dangling DNS record is considered hijackable. Therefore, as a security best practice, a DNS record should be purged from its corresponding DNS zone once it becomes dangling.

Unfortunately, in practice, domain owners often forget to do the cleaning, resulting in many dangling DNS records.

There are tens of DNS record types in DNS specifications, several of which could lead to hijackable dangling records. In this blog, we focus on three types of records, as listed in Table 1.

Type	Description
CNAME	Indicate the rrname is an alias for the canonical name rdata.
MX	Specify the mail server responsible for accepting emails on behalf of the domain.
NS	Delegate to an authoritative name server.

Table 1. Types of dangling DNS records studied here.

A CNAME record specifies the canonical name (rdata) of an alias domain (rrname). A DNS query for the alias will be resolved to its canonical name, which is further resolved to an A/AAAA record. A dangling CNAME record could result in the domain name in rrname being hijacked by attackers.

For instance, in the following CNAME record, `dangling.example[.]com` points to an expired domain. Therefore, this CNAME record should be purged from the zone file. Otherwise, `dangling.example[.]com` will be hijacked if the expired domain expired[.]com is registered by an attacker.

```
dangling.example[.]com CNAME expired[.]com
```

An MX record specifies the mail server responsible for receiving emails on behalf of the domain in rrname. A domain can have multiple MX records with different priorities. The MX record with higher priority will be used first. A hijacked dangling MX record allows attackers to send and receive emails under the affected domain in rrname.

An NS record delegates a domain (rrname) to an authoritative DNS name server (aDNS) to answer queries about names under that domain. If a dangling NS record is exploited, attackers will be able to control the aDNS and redirect all domain visitors to any IP address. Even worse, the transitive trust that is built into DNS could make all domains that directly or indirectly depend on the dangling NS record vulnerable. A domain usually has multiple NS records, and DNS resolvers can use different algorithms to decide which NS record to use. When only one of the multiple NS records is dangling, attackers can leverage techniques like denial of service and NS pinning to force DNS resolvers to use the dangling NS record.

Dangling Domain Hijacking

A dangling record is safe as long as the abandoned resource cannot be manipulated by anyone other than the domain owner. Otherwise, it is an unsafe dangling record. A previous study, "All Your DNS Records Point to Us," has published multiple methods that can be used to exploit dangling records. Here, we describe two of them for dangling records for which the rdata is a domain.

The first method can be used if the rdata in all of the three DNS record types in Table 1 is a domain name. The domain in rdata could expire

and thus could be re-registered by attackers. Registering expired rdata is significantly different from previous attacks that exploit the residual trust of the expired domains themselves. In dangling domain hijacking, the attackers instead abuse the trust of the unexpired rrname. There are several reasons why domain owners frequently neglect such dangling records. One reason is that there are usually multiple MX/NS records, and some are still working. Thus, the services depending on these MX/NS records are usually not interrupted. Another common reason is that the services pointed to by the dangling records are no longer used and no one bothers to update the record.

The second method takes advantage of abandoned third-party services and is sketched in Figure 1. Third-party services are extensively used in modern websites. For instance, GitHub and WordPress are widely used to build home pages using the virtual hosting technique. Users add a DNS record in their DNS zone file pointing custom domains to the domains or IP addresses where the services are hosted.

For example, a user wants to host a home page using WordPress on the domain `blog.mydom[.]com`. The user needs to add the following record to the zone file of `mydom[.]com`:

```
blog.mydom[.]com CNAME example.wordpress[.]com
```

where example.wordpress[.]com is a domain name allocated by WordPress. Alternatively, an A record could be added instead of the CNAME record:

```
blog.mydom[.]com A 192[.]0.78.24
```

Then, the user's WordPress account can claim `blog.mydom[.]com` so that all visits to the domain will be directed to the website set up on WordPress using the virtual hosting technique. Later, when the user does not want to use WordPress anymore, `blog.mydom[.]com` may be unclaimed in the user's WordPress account. If the above `CNAME/A` DNS record is not removed from the zone file of mydom.com, now the record becomes dangling. To exploit the dangling record, an attacker simply needs to register a WordPress account and then claim the ownership of `blog.mydom[.]com` in the attacker's WordPress account. Although the attacker does not own `example.wordpress[.]com`, all visits to `blog.mydom[.]com` will still be directed to the website set up by the attacker on WordPress. Actually, the domain `example.wordpress[.]com` in rdata does not matter here because virtual hosting uses the Host field (i.e. `blog.mydom[.]com`) in HTTP requests to serve different websites. That is why a user can also use an A record to point to the same IP address owned by WordPress.

The same risk applies to GitHub, as stated in their tutorial – "*Make sure you add your custom domain to your GitHub Pages site before configuring your custom domain with your DNS provider. Configuring your custom domain with your DNS provider without adding your custom domain to GitHub could result in someone else being able to host a site on one of your subdomains.*"

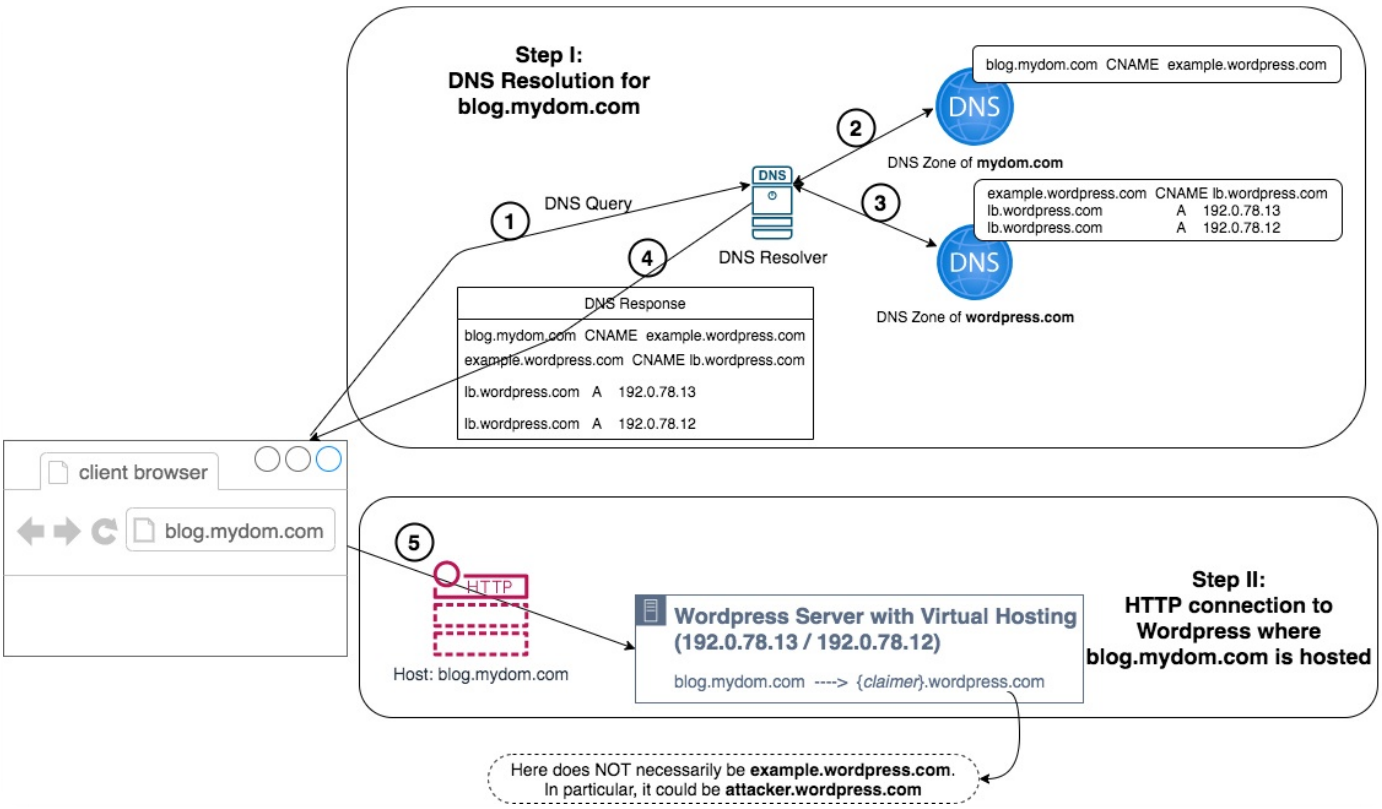


Figure 1. Illustration of the workflow when a client browser visits a domain hosted on WordPress.

Dangling domain hijacking can significantly increase the efficacy of scamming and phishing. A common modus operandi for attackers is to register a domain with a benign reputation or to use a domain similar to well-known legitimate ones (as seen with squatting domains). However, these techniques are limited in efficacy, and vigilant users can easily spot them. Moreover, many automatic systems such as proactive malicious NRD detector and squatting domain detector have been deployed in production to detect these malicious domains.

With dangling domain hijacking, on the other hand, attackers can easily abuse domains with clean history at an affordable cost. In particular, the whois information of these hijacked dangling domains remains the same. Even worse, dangling subdomains often inherit the reputation of their parent domains, some of which are well-known and have good reputations. Therefore, it is critical to protect users against unsafe dangling domains.

Dangling Domain Detection

Once we know the techniques used to hijack dangling domains, it is straightforward to implement the detector. For a domain in rdata, we check whether the domain has expired. If so, the DNS record is dangling. If the rdata indicates a third-party service, we check whether the rrname can be claimed in the corresponding third-party service. Since a domain can become dangling at any time, we need to periodically check every valid DNS record. Therefore, our detector checks every valid DNS record in our passive DNS every few weeks. Meanwhile, to protect users in a timely fashion, we also conduct daily detection for all actively queried DNS records in the past day.

Prevalence of Dangling Domains

With our dangling domain detector, we have detected 317,000 unsafe dangling domains in total. Figure 2 shows the breakdown of dangling domain types, with 63.1% being expired rdata, 36.9% from GitHub and 0.1% from WordPress. In addition, we show their distribution of DNS record types in Figure 3, revealing that the vast majority of these records are CNAME. Surprisingly, we did not detect dangling MX records. There are two possible reasons for this. First, most MX records in the wild point to third-party mailing services such as Mailgun. Second, according to the previous study, it is rare in practice for expired rdata to result in dangling MX records.

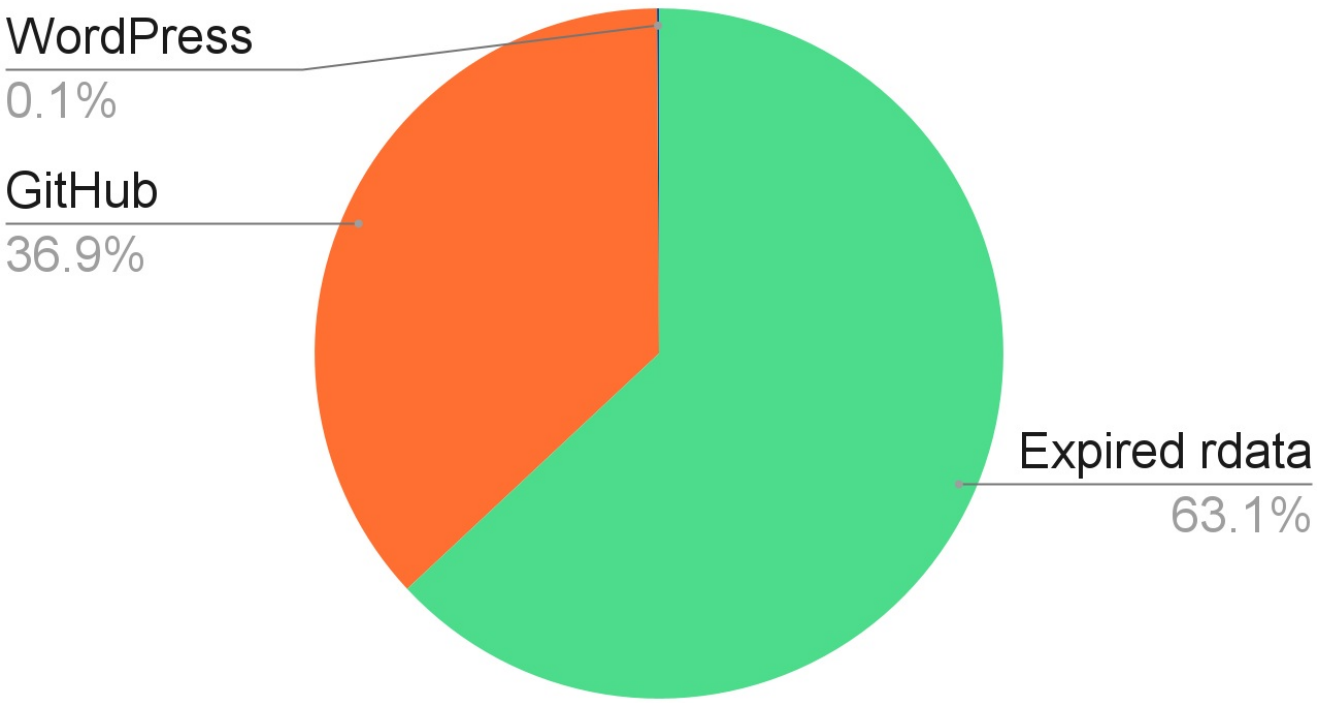


Figure 2. Breakdown of dangling domain types.

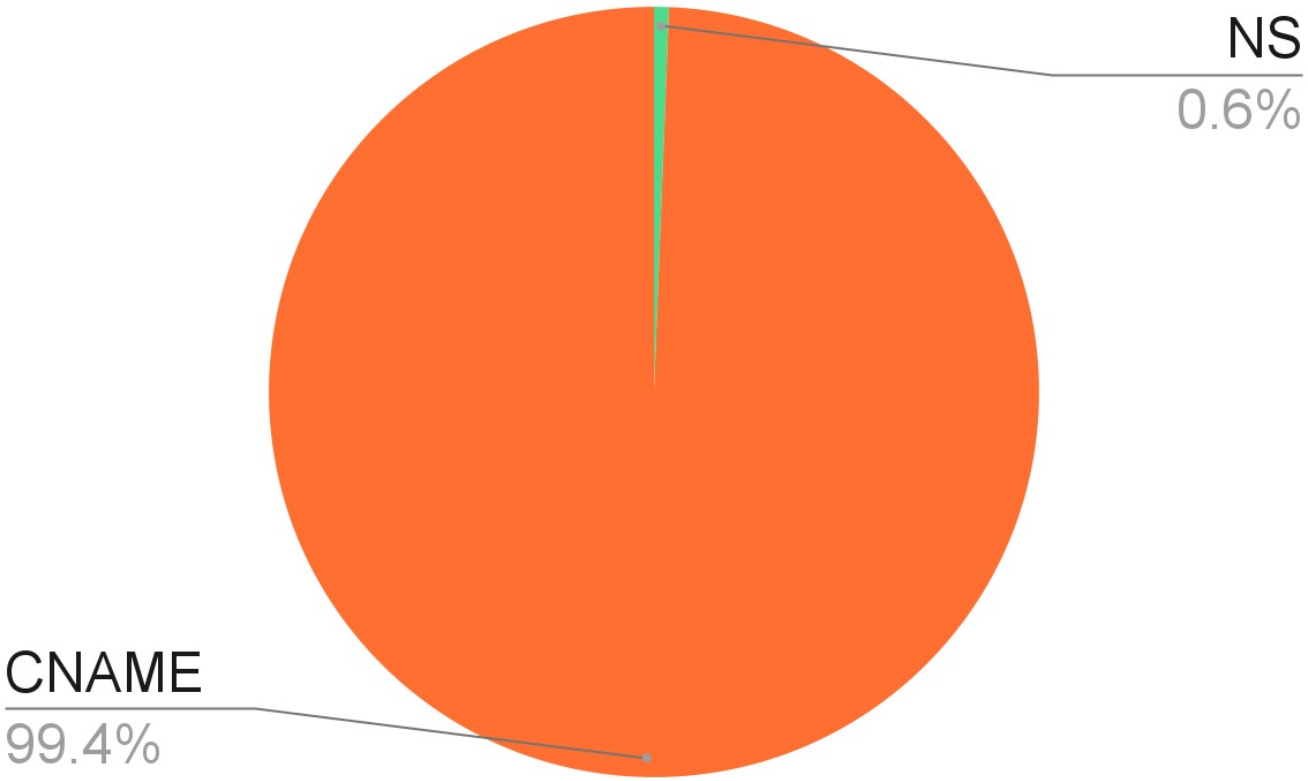


Figure 3. Distribution of DNS record type.

To understand the prevalence of dangling domains, we checked the passive DNS dataset from May 22-June 21, to see how many are actively queried. We excluded 88 wildcard dangling domains because matching them is computationally expensive. The number of unique dangling domains resolved each day is presented in Figure 4. It shows that several thousand dangling domains are queried daily. There are two spikes on 2021-06-09 and 2021-06-12 in Figure 4, and further analysis shows that the spikes are caused by a single domain, which corresponds to over 11,000 unique dangling subdomains on the two days.

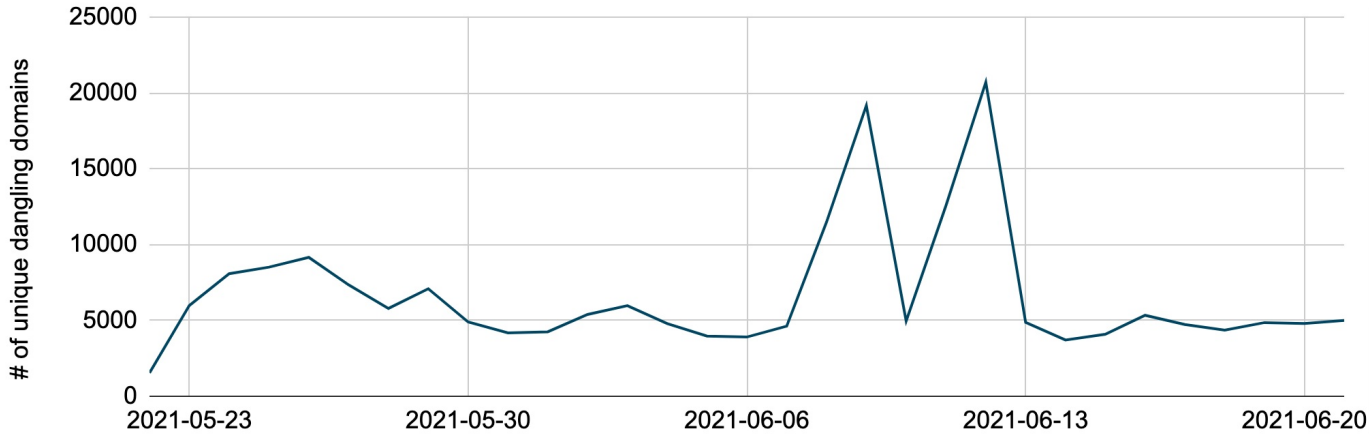


Figure 4. The number of unique dangling domains resolved each day from May 22-June 21.

To understand the risks of these dangling domains, we aggregated the 317,000 dangling domains by TLD and presented the top 60 TLDs in Figure 5. The top TLD is `com`, which accounts for 55.2% of all dangling domains. Of particular interest, the TLDs `edu` and `gov` are considered well-managed DNS zones (adhering to eligibility requirements and strict process for registering new domains), but they still have 197 and 13 dangling domains, respectively. As a result, attackers could exploit the trust in edu and gov for attacks like phishing and scams. Additionally, we checked these dangling domains to see if they are subdomains of Tranco's top 1 million domains, which implies inherited trust. The results show that 38,000 (12.0%) of them are under Tranco top 1 million domains. We present the distribution of their Tranco rank in Figure 6. In particular, 4,767 of them are under the Tranco top 2,000.

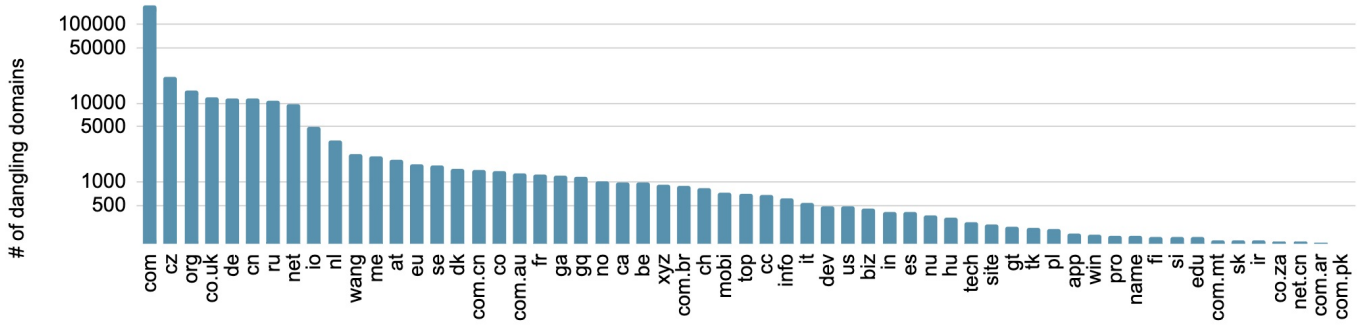


Figure 5. The number of dangling domains aggregated by TLD.

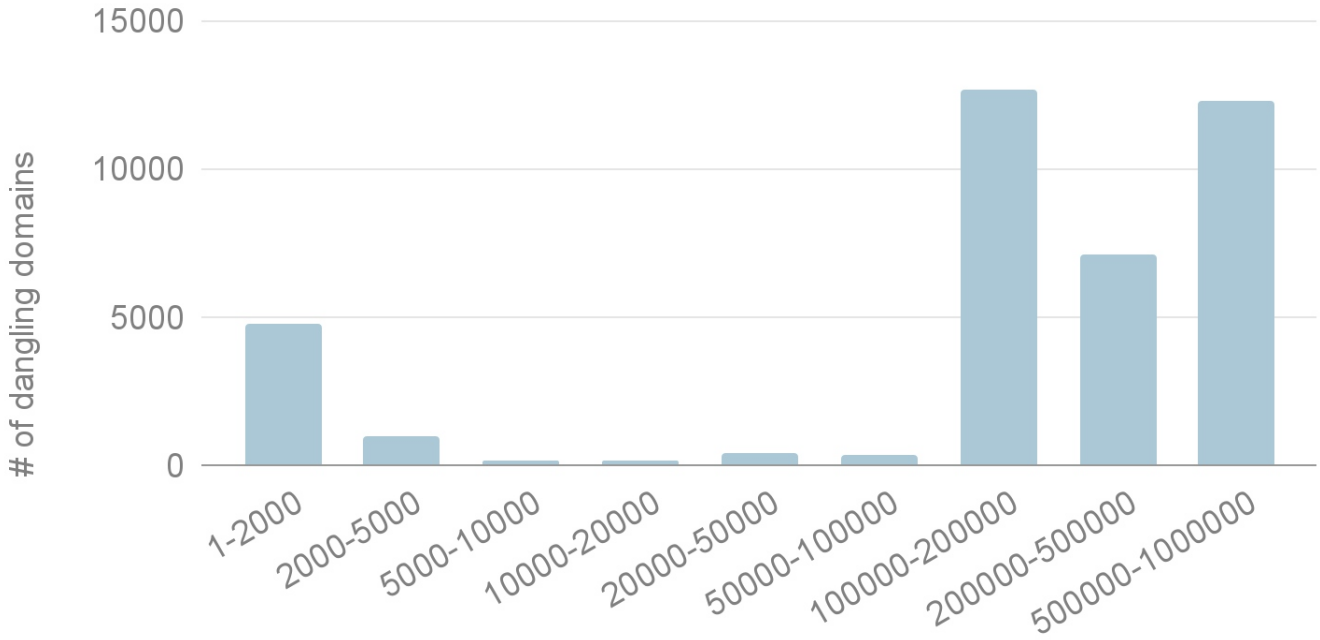


Figure 6. The Tranco rank distribution for the 38,000 domains that are subdomains of Tranco top 1 million domains.

Case Studies

During our evaluation of the detection results, we observed many compelling cases and patterns. In this section, we describe several representative instances.

Subdomains Inherit Trust From High-Profile Parents

Due to the hierarchical structure of the DNS system, subdomains usually inherit trust from their parents, and a subdomain is often given the same credibility as its parent. For instance, the web contents hosted on a subdomain under edu are considered official information from colleges or universities. This makes dangling domains under reputable domains quite attractive to attackers. During our study, we spotted multiple such cases. Two examples are shown in Figure 7.

admitted.gradengineering.[redacted].edu CNAME www.[redacted]gradengineering.info

access.publichealth.[redacted].edu CNAME kc-whitelabel-portal.[redacted]gts.com

Figure 7. Two examples of dangling domains under edu.

The two domains in rrname are owned by two well-known universities in the U.S. Visitors to the websites hosted on these domains are likely to consider the information as valid and official from the universities. As a result, if an attacker hosts malicious content or carries out spear phishing under such subdomains, even the most vigilant users could fall victim.

Typos Cause Dangling DNS Records

Typos are a common cause of dangling domains. We found many dangling records where rdata is obviously a typo. For example, in Figure 8, the CNAME DNS record, `cdcr.[xxxxx.]edu` points to a subdomain under `cloudlfare[.]net` which has expired at the time of writing. We speculate with high confidence that `cloudlfare[.]net` is a typo of cloudflare.net, which provides a content delivery network (CDN) and other web-related services.

Figure 8. Typos are a common cause of dangling domains, as shown in the example here.

We dug deeper to figure out why the administrator of `cdcr.[xxxx.]edu` did not notice this error. First, according to information we found online, the dangling domain `cdcr.[xxxx.]edu` is probably the home page for one of the university's communities. Our passive DNS data shows that the first time the dangling DNS record was seen was Oct. 17, 2017. Before this date, `cdcr.[xxxx.]edu` points to `webhost302.[xxxx.]edu` using the CNAME record. This suggests that the domain administrator intended to switch to Cloudflare on Oct. 17, 2017. Then, we extracted the WHOIS data of `cloudlfare[.]net`, as shown in Figure 9. We can see that the domain was registered on Dec. 7, 2017, about two months later, after the dangling DNS record was created. Our passive DNS data of `cloudlfare[.]net` matches its registration date. This implies that `cloudlfare[.]net` was probably not registered at the time `cdcr.[xxxx.]edu` started to point to `cdcr.[xxxx.]edu.cdn.cloudlfare[.]net`, and the DNS record was dangling at the time of creation. In summary, it remains unknown why the administrator of `cdcr.[xxxx.]edu` did not notice this error.

Figure 9. WHOIS data of cloudlfare[.]net.

Dependency on Expired Services

As we mentioned above, a significant number of domains point to third-party services. These third-party services could discontinue or migrate to a new domain to carry out services. In such cases, the original domains could expire, and thus all domains pointing to them become dangling. Take the DNS record in Figure 10 as an example.

Figure 10. The example shows how a domain can become dangling when it points to a third-party service that has been discontinued or has migrated to a new domain.

According to slides still hosted online, `getwebsitesearch[.]com` was owned by a startup that provided custom search engines to websites. However, the company has discontinued its service, and thus the domain has expired.

Dangling Wildcard DNS Record

Most dangling DNS records affect only a single domain. However, we found 88 wildcard DNS records that became dangling in our passive

DNS. Such dangling records are quite interesting in that all nonexistent domains under the related rrname could be hijacked by attackers. For example, because of the dangling record shown in Figure 11, all domains not explicitly specified in the zone file of `ch[xxxxxxx]wer[.]com` could be hijacked. In particular, attackers could spawn an infinite number of subdomains for malicious purposes such as hosting phishing content and acting as command and control. Even worse, the presence of dangling wildcard DNS records makes it more difficult for defenders to block the hijacked domains. The only reliable defense is to remove the dangling record or defensively take over the wildcard domain.

Figure 11. An example of a dangling wildcard DNS record.

Dangling NS Record

As described above, a dangling NS record could render all domains delegated to it hijackable. Therefore, we checked how many dangling domains are detected in our data set. In total, we found 1,974 dangling NS records under 1,659 unique root domains. Note that we excluded the DNS records for which the root domains of rrname and rdata are the same. The DNS records where the rrname has expired were also excluded. Interestingly, we found that many rrnames are delegated to a single expired name server domain. For instance, 15 unique rrnames with different root domains are delegated to the same name server `ns.a.cloudtabo[.]com` and `ns.b.cloudtabo[.]com`. As a result, the attacker just needs to control a single domain to hijack 15 others. We manually checked these 15 rrnames and found that they all have redundant NS records pointing to name servers `ns.c.clouddra[.]com` and `ns.d.clouddra[.]com`. Since `clouddra[.]com` is still valid, the affected 15 domains can still work properly. However, attackers can still hijack partial traffic to these 15 domains and potentially take full control by leveraging denial-of-service and NS pinning techniques.

Conclusion

This blog has introduced the concept of dangling DNS records and demonstrated that dangling domains are still prevalent and can pose serious security threats. In particular, we have found 317,000 dangling domains, including thousands on high-profile DNS zones such as `edu`, `gov` and Tranco top.

Palo Alto Networks identifies the detected dangling domains with the grayware category through our security subscriptions for Next-Generation Firewalls, including DNS Security and Advanced URL Filtering. Our customers are protected against damage from the risky domains mentioned above as well as against other risky domains captured by our system.

Palo Alto Networks has notified the owners of the detected dangling domains. The dangling domains mentioned in this blog have been defensively taken over and are no longer serving malicious content.

Additional Resources

- DNS Rebinding Attack: How Malicious Websites Exploit Private Networks
- Detecting and Preventing Malicious Domains Proactively with DNS Security
- Fast Flux 101: How Cybercriminals Improve the Resilience of Their Infrastructure to Evade Detection and Law Enforcement Takedowns
- The History of DNS Vulnerabilities and the Cloud