2023 LONDON

4 - 6 October, 2023 / London, United Kingdom

‘DIGITAL ROADS TO NOWHERE’: A RE-ASSESSMENT OF DANGLING DNS RECORDS, AND SUBDOMAIN TAKEOVERS

John Jensen, Ken Bagnall & Gareth Howells

Silent Push, USA

jjensen@silentpush.com
kbagnall@silentpush.com
ghowells@silentpush.com

ABSTRACT

This study reconsiders the practice of DNS record enumeration within the context of dangling DNS records, by arguing that contemporary brute-force enumeration techniques are no longer adequate to deal with the growing problem of subdomain takeovers. Throughout the study, we refer to proprietary DNS scanning software that scans and logs each DNS record available on the internet's IPv4 scope, and places them in a database solution that can be used at scale. We explore how DNS records become dangling, and what the consequences are for organizations by categorizing the risks per record type, and presenting real-world case studies that show how records are manipulated by threat actors. We then make recommendations on how organizations and security vendors can improve their public DNS security through the adoption of recently developed technology and more robust DNS housekeeping.

INTRODUCTION

According to its creator, the US computer scientist Paul Mockapetris, DNS was developed in 1983 to 'accommodate diversity without unnecessary restriction' without any native security features [1]. Removing the operational and financial barriers to creating internet-based infrastructure was necessary to facilitate the expansion of another technological marvel, the World Wide Web, slowly but surely making its way out of a CERN laboratory in the late '80s.

Parkinson's law – the notion that administrative systems expand regardless of the amount of work required to maintain efficiency – is not limited to paper-based scenarios. Fast forward 40 years and the public DNS landscape has evolved into an ingenious but inexact trust-based system that – whilst maintaining its central role in expansion of the global commerce and fundamentally changing the way human beings communicate with each other – features weaknesses hiding in plain sight, that can lead to profound financial and reputational damage for any organization that fails to acknowledge them.

This study uses modern DNS enumeration techniques and real-world case studies to analyse the anatomy of subdomain takeovers caused by dangling DNS records – DNS entries pointing at non-existent or uncontrolled resources – challenging the assumption that the non-existence of resources is in any way secure, and suggest ways in which the security community can better protect its customers and user base by being smarter with the data and tools at hand.

BACKGROUND

Before attempting to understand the contemporary problem of dangling DNS exploitation, it is first necessary to understand how things came to be. The issue of subdomain takeovers is symptomatic of the overall human approach to computational system design. Distributed systems, be they cloud-native or otherwise, struggle to deal with digital roads to nowhere. Classical systems design theory encourages a place for everything, and everything in its place. DNS is 'unduly burdened' with the demands of the modern-day internet [2]. In the context of subdomains, the problem is immediately apparent – DNS records and web pages do not administer themselves. Organizations are charged with maintaining an accurate, up-to-date, and operationally sound public DNS inventory that should only contain records which facilitate connections to a controllable and secure set of resources – or at the very least, resources that actually exist.

Recent treatment

As far back as 2010, throughout the advancement of SaaS, PaaS, IaaS or any other 'as-a-service' operating models that caused an exponential explosion in web-based resources (and the need for subdomains to host them), prominent computer scientists were acknowledging the need to build web-centric systems that were mindful of pointer-based drawbacks, and took steps to address them.

In their 2010 book *REST in Practice: Hypermedia and Systems Architecture*, Jim Webber, Savas Parastatidis and Ian Robinson use the so-called 'heresy' of Error 404 as an example of how human beings are unable to exercise full control of the internet, all of the time [3]. Whilst this may be true for the internet as a whole, the opposite is true of self-contained organizational DNS systems. Complete control in this case is a distinct possibility, as we shall see.

The first dedicated and detailed study of subdomain takeovers was presented at the 2016 ACM SIGSAC Conference on Computer and Communications Security in Vienna, by researchers from the University of Delaware and the College of William and Mary, Virginia. In their influential paper, *All Your DNS Records Point to Us: Understanding the Security Threats of Dangling DNS Records,* Daiping Liu, Shuai Hao and Haining Wang [4] exposed and picked apart the intertwined concepts of dangling DNS records and subdomain takeovers – a form of DNS-based attack vector that was widely overlooked by the security community at the time. Their conclusion – that dangling DNS records are mostly a computational problem, rather than a human one – is, however, problematic. The topic of poor DNS housekeeping will occur frequently throughout this study and is impossible to ignore as the primary correlative factor. Having said that, we should bear in mind that Liu, Hao and Wang were operating with a limited set of tools and resources. If a year is a long time in politics, then seven years is an eon in cybersecurity, and the industry has developed far more robust methods of enumerating DNS records, as we shall see.

Modern sentiments

Since the 2016 SIGSAC study, academia has given the topic a great deal of attention, but private sector DNS subdomain defences are still not where they need to be. A cursory search through academic repositories returns a multitude of papers published from 2016 onwards dealing with the subject of dangling DNS and subdomain takeovers, from a range of conceptual and practical angles.

Unfortunately, the subject's new-found notoriety has not been the catalyst for real-world reform that many in the industry had hoped for, due to inadequate enumeration tooling and poor DNS housekeeping. In 2022, industry research suggested that within a sample pool of apex domains between 2020 and 2021, subdomain vulnerabilities increased as much as 25% [5]. Organizations are either not being provided with the correct consultative advice on how to bolster their public DNS defences, or are wilfully choosing to forego subdomain-related cyber defence mechanisms for financial and operational reasons. In the absence of authoritative research, it is likely a combination of both.

WHAT ARE DANGLING DNS RECORDS AND SUBDOMAIN TAKEOVERS?

Dangling DNS records are DNS records that point to a deprovisioned, abandoned or otherwise uncontrolled resource (services, IP address etc.), which can be used by threat actors to propagate exploits across the apex domain – also known as a subdomain takeover.

DNS record type	Function
`CNAME`	Points to `A/AAAA` record or another `CNAME`
`NS`	Points to `A/AAAA` record
`MX`	Points to `A/AAAA` record
`Glue`	Points to `NS` records
`SPF Include`	Points to `A/AAAA/MX/SPF` records
`A`	Points to an IP outside of ownership zone
`AAAA`	Points to an IP outside of ownership zone

Table 1: Types of dangling DNS records.

Given that they are the record targeted most by threat actors, let's look at a common scenario featuring a dangling `CNAME` record. Later in the study, we'll delve into some real-world examples.

An organization wants to host an e-commerce page provided by a third-party vendor on the domain `shopname.mydomain[.]com`. They add a `CNAME` record to the DNS records of `mydomain[.]com`, that points to a resource at `shopname.ecommerce-site[.]com`. The brand chooses to move to another e-commerce provider and `shopname.ecommerce-site[.]com` falls out of use, which means that the `CNAME` entry in the zone file of `mydomain[.]com` is now pointing to a non-existent resource. The brand fails to delete the accompanying `CNAME` entry, which becomes a 'dangling' record. An attacker initiates a subdomain takeover by creating an account at `ecommerce-site[.]com` and re-initializing the decommissioned resource at `shopname.ecommerce-site[.]com`. Given that, in our scenario, the corresponding `CNAME` has not been removed, traffic to the subdomain at `shopname.mydomain[.]com` will now be directed to an attacker-controlled page, as well as allowing the hacker to wreak further havoc across the DNS scope of the apex domain.

Example scenario – negligent registrar behaviour

> *Note: All case studies are real-world examples of dangling DNS records and subdomain takeovers. The domains in question were temporarily taken over by Silent Push as proof of concept and to prevent seizure by malicious groups. No network infiltration occurred, and a holding page was displayed, asking domain owners to contact Silent Push.*

Although some types are more susceptible than others, any DNS record that points to misconfigured or absent resources is susceptible to manipulation, and the problem is not limited to non-tech companies who administer their own public DNS presence. Let's look at a real-world example involving an authoritative nameserver that points to the wrong set of registrar `NS` records.

Figure 1 shows information for the domain `actimmi.com[.]au`, with three associated nameservers registered by the registrar, `melbourneit[.]com`.

```
whois actimmi.com.au
Domain name: ACTIMMI.COM.AU
Registrar WHOIS Server: whois.auda.org.au
Registrar URL: https://www.melbourneit.com.au/contact-us/
Name Server: NS1.MELBOURNEIT.NET
Name Server: NS2.MELBOURNEIT.NET
Name Server: NS3.MELBOURNEIT.NET
```

Figure 1: Melbourne IT WHOIS information.

However, when using a *Linux* `dig` command to query the authoritative nameservers of `actimmi.com[.]au`, we discovered that the registrar missed the 'e' from the end of `melbourne`:

```
dig @ns1.melbourneit.net ns actimmi.com.au

actimmi.com.au 3600 IN NS ns3.melbournit.net
actimmi.com.au 3600 IN NS ns4.melbournit.net
actimmi.com.au 3600 IN NS ns1.melbournit.net
actimmi.com.au 3600 IN NS ns2.melbournit.net
```

Figure 2: Incorrect nameserver information for actimmi.com.au.

The outcome of the misspelt domain name is that `actimmi.com[.]au` was delegated to a set of nameservers that were not controlled by the host organization or the registrar (`nsx.melbournit.net`).

Threat actors would have been able to register `melbournit[.]com` (which was available), set up the misspelt nameservers and respond with malicious content to requests for any record in `actimmi.com[.]au`, once DNS caching is no longer a factor, including phishing websites within the target's own domain.

Thankfully in this case, we were able to register `melbournit[.]com` and alert both `actimmi.com[.]au` and the registrar to the misspelt nameserver.

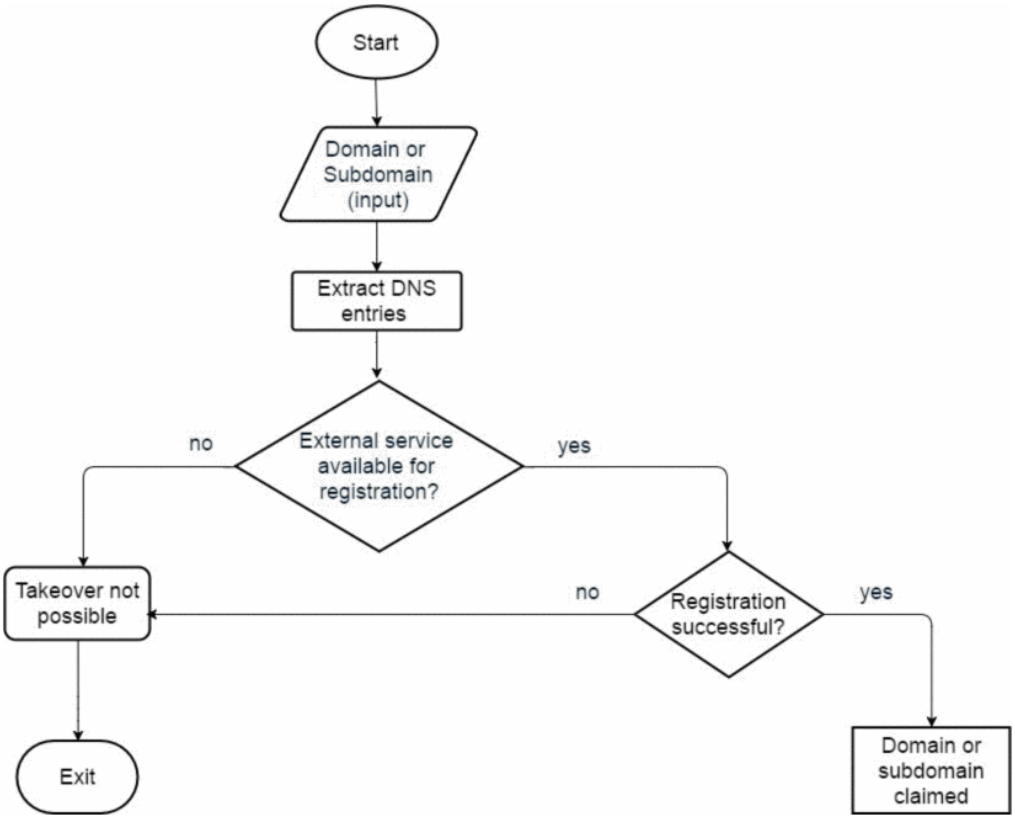

Figure 3: Subdomain takeover process. Source: Rashid et al. [6].

Consequences of an unresolved dangling record

Subdomain takeovers can lead to all manner of direct and indirect reputational, operational, and financial consequences. Once a subdomain has been claimed, attackers are able to quickly establish web pages that host malicious content and propagate a variety of exploits across the scope of the apex domain, including nameserver hijacks, phishing attacks, authentication exploits, XSS attacks and email spoofing.

For example, if an organization shares browser cookies across some or all of their subdomains, and one of those subdomains is hijacked, they run the risk of not only allowing a threat actor to utilize hashed credentials stored in the cookie and authenticate themselves as a user, but exposing the company-wide SSO service, and all the internal systems and external systems that it provides access to.

Potential outcomes per record type

Most of the discussion surrounding dangling DNS records relates primarily to `CNAME` records, but the concept applies to a multitude of DNS record types that point to a resource that relates back to the apex zone file but is *outside* of an organization's control.

Dangling record type	Consequences
`CNAME`	Takeover of target website. Cookie harvesting and browser data. SSO authentication compromised. Malicious content served on organizational domains.
`NS`	Total or partial takeover of domain traffic, depending on the number of nameservers compromised. Redirection of DNS records within the domain zone.
`MX`	Total or partial takeover of email traffic. Email spoofing. SPF impersonation.
`Glue`	Total takeover of domain zone.
`TXT/SPF`	Email spoofing
`A/AAAA`	Takeover of target website or host.

Table 2: The consequences of subdomain takeovers per record type.

Along with `CNAME` entries, `NS` and `MX` records are the most common type of dangling record found across the internet's IPv4 space, comprising the 'big three' sub-group of exploitable records (see Table 3). Common exploits include the hosting of malicious content, nameserver hijacking, SSO exploitation and cookie harvesting.

A domain's `Glue` records, used to resolve circular DNS dependencies, point to the IP addresses of a domain's nameservers. Threat actors exploit `Glue` records by establishing a DNS server with the same IP address as the legitimate nameserver. When a user attempts to resolve the subdomain, their DNS resolver will send the request to the rogue server, which will then return the IP address of the attacker's server, facilitating a malicious redirect.

`TXT` records contain Sender Policy Framework (SPF) instructions that are used to specify which domain names and IP addresses are authorized to send email on behalf of an apex domain. Public cloud vendors and email providers sometimes operate with a set of default SPF instructions that they recommend network administrators add to a zone file to facilitate a particular service. These embedded IP ranges can number in the thousands, or hundreds of thousands, exposing an apex domain to manipulation from an innumerable number of sources.

Similarly, misconfigured `A` or `AAAA` records pointing to an IP address that is no longer controlled by the apex domain – as happens during the provisioning and decommissioning of VMs and public cloud infrastructure – are equally exploitable.

LOCATING DANGLING DNS RECORDS

Given that we know how dangerous dangling DNS records are, and what they can lead to, the question presents itself: how are organizational DNS records discovered, from a defender's point of view, to prevent attacks? The answer is – without smart application of technology – with considerable and unnecessary difficulty.

Manual location

The 'manual' method is to use a lookup service to verify that individual domains and subdomains translate to an IP address – whether directly, or in a chain. In Figure 4 we see the DNS configuration of `microsoft[.]com` passing through three `CNAME` records before resolving to an `A` record with an associated IP (`23.50.125[.]163`).

```
dig www.microsoft.com

www.microsoft.com                                   1632    IN    CNAME   www.microsoft.com-c-3.edgekey.net
www.microsoft.com-c-3.edgekey.net                   137     IN    CNAME   www.microsoft.com-c-3.edgekey.net.globalredir.akadns.net
www.microsoft.com-c-3.edgekey.net.globalredir.akadns.net  137   IN    CNAME   e13678.dscb.akamaiedge.net
e13678.dscb.akamaiedge.net                          16      IN    A       23.50.125.163
```

Figure 4: DNS resolution for www.microsoft[.]com.

Here is another example showing a dangling `CNAME` entry for `www.sc.microsoft[.]com`, with no associated IP:

```
dig www.sc.microsoft.com

www.sc.microsoft.com    3600    IN    CNAME   sc.microsoft.com
```

Figure 5: Dangling CNAME record for www.sc.microsoft[.]com.

Classical pseudo-brute force lookups, such as the above example, are simply not feasible to use as a defence mechanism at any kind of scale. `CNAME` records do not exist on the apex domain, they are only applicable towards subdomains and hostnames. If we take the approximate amount of apex domains in existence on the IPv4 space at any given time to be anywhere between 500 and 600 million, we then need to multiply this by the number of `CNAME` entries available to each domain to discover the total number of records to be scanned for exploits.

Automated dangling DNS scanning

How, then, are organizations expected to cope, given that reliable scanning of global DNS records is not possible with manual searching or even partial automation. In recent years, *Silent Push* has developed a proprietary IPv4 and DNS scanning algorithm that combats the inherent inefficiencies of brute force lookups. Instead of looking for the figurative needle (record) in a haystack (IPv4 range), custom DNS record stacks contain *only needles.*

To quote a certain Victorian era crime fighter who understood a thing or two about logical reasoning: 'Once you eliminate the impossible, whatever remains, no matter how improbable, must be the truth.' The *Silent Push* algorithm resolves every public DNS record available on the internet daily, encompassing many billions of records, producing a continuously updated global record set using active DNS scanning.

The collection method then morphs into a database solution that explores the partitioned symmetric difference between record types to enumerate dangling `CNAME` records (or whatever record is required in the lookup). This is achieved by isolating all `CNAME` records within the record set before subtracting all the target `A`/`AAAA` records. What we are then left with is an enumerated list of `CNAME` records with no target.

In an ideal world, the outcome of that difference would be an empty set, given that all `CNAME` records should be mapped to `A`/`AAAA` records. Unfortunately, digital roads to nowhere do exist. Not in the hundreds. Not even in the thousands, or tens of thousands, but in the *millions.*

Quantifying the number of dangling DNS records on the internet

Table 3 shows almost 9 million dangling DNS records per record type, collected from the internet's entire IPv4 space. This dataset is what a threat actor would undoubtedly deem a 'target-rich environment'. Reliable attacker-side enumeration allows threat groups to re-focus on weak infrastructure and be a lot more efficient with the time and resources they have available. At present, the main (and only) barrier to a dramatic global increase in phishing attacks and subdomain takeovers is an attacker's ability to discover dangling records available for exploit in a timely and resource/cost-efficient manner. Once this hurdle has been removed, all bets are well and truly off.

Record type	Number of dangling records
`CNAME`	5,201,101
`NS`	329,465
`MX`	3,514,389

Table 3: Estimated number of dangling records following a full IPv4 scan.

The problem of dangling DNS records is not limited to micro-businesses, start-ups or SMBs with small security budgets and a lack of governance. Last year, researchers at *Silent Push* detected 70 expired services on *Microsoft Azure*'s content delivery network (`azureedge[.]net`) with an attached domain, that run the risk of being hijacked, and 80 similar records across *GitHub* [7].

Mining dangling DNS records from automated scans

Whilst advanced enumeration opens a world of opportunity for threat actors, commercially available white hat tools represent a significant step forward in the battle against dangling DNS records and subdomain takeovers. Private cybersecurity investment capital needs to acknowledge the urgent need for algorithmic, database-driven approaches to enumeration, and offer organizations a means to protect their public DNS infrastructure through intuitive, easily accessible global DNS lookup tooling.

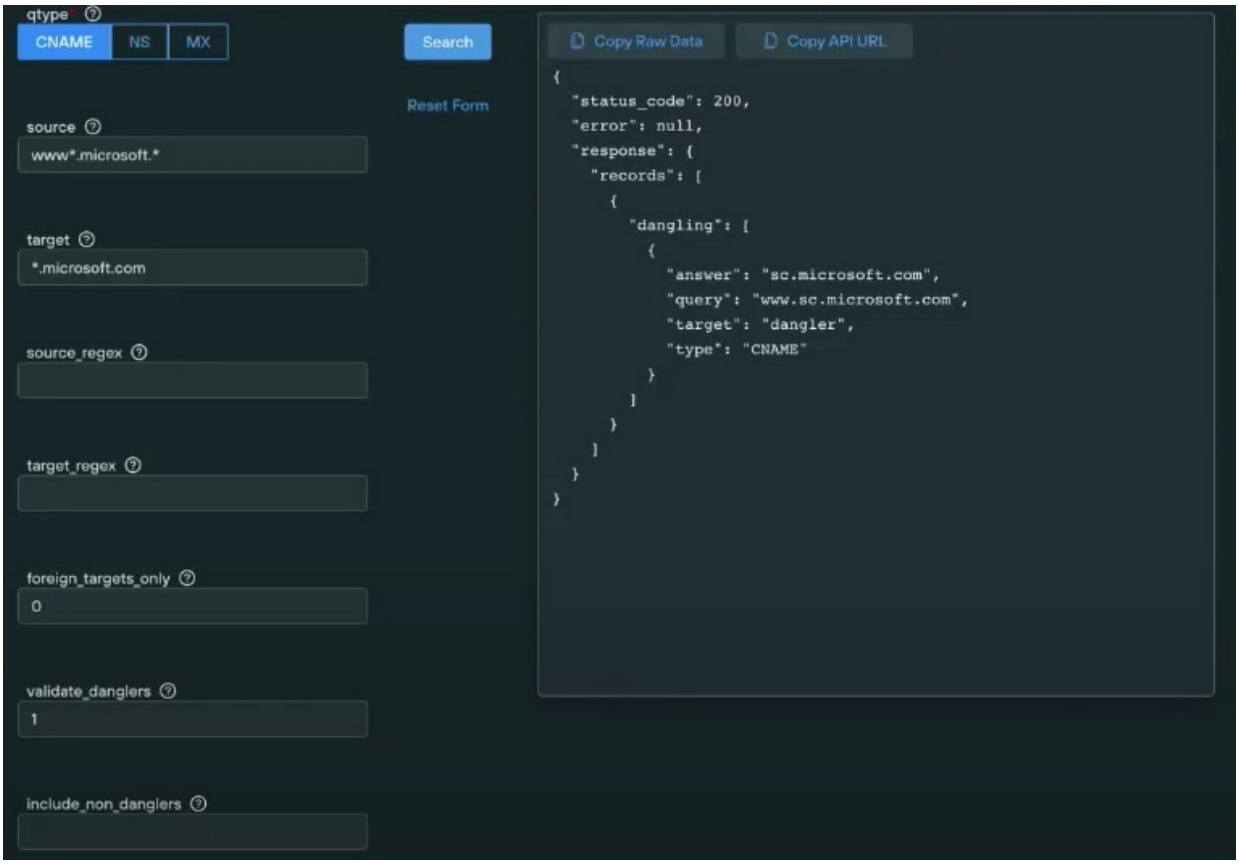

Figure 6: Mining dangling DNS records in a GUI using Silent Push.

CASE STUDY – THE ANATOMY OF A CNAME TAKEOVER

What makes certain subdomain-based attack vectors so difficult to identify in the eyes of a casual site visitor lies in the domain itself. Rather than threat actors relying upon overt *apex* domain 'typosquatting' (identified by Kintis et al. as 'combosquatting') attack vectors that exploit subtle differences in the alphanumeric name of root domain URLs and populate landing pages with malicious content, compromised *subdomains* often appear as legitimate constituent parts of an organization's root domain – and from a domain traffic perspective, are just that [8].

Let's take a detailed look at the anatomy of a subdomain takeover, propagated by misconfigured DNS infrastructure. Oxfam is a large British charity that generated over £300 million through donations, trading and investments between 2021 and 2022 [9]. Every year, the organization runs a series of live music events across the UK called Oxjam that generate a significant amount of revenue.

Step 1 – Discovery of dangling CNAME

In July 2022, we discovered that a `CNAME` record pointing to `www.oxfamhosted.co[.]uk` from `www.oxjam[.]org` was dangling by using the aforementioned enumeration algorithm, compromising the domain infrastructure of one of the UK's largest charitable organizations. The target website for the `CNAME` record, `www.oxfamhosted.co[.]uk`, did not exist.

```
dig www.oxjam.org

www.oxjam.org   21600   IN    CNAME   www.oxfamhosted.co.uk
```

Figure 7: Dangling CNAME record for oxfamhosted[.]co.uk.

Step 2 – Domain acquisition

We purchased `www.oxfamhosted.co[.]uk` for $2, allowing the team to configure a DNS server of their own.

Step 3 – DNS re-pointing

Once the domain was acquired, we were able to resolve the DNS name chain to our own webserver at 49.x.x.x from `www.oxjam[.]org`:

```
dig www.oxfamhosted.co.uk

www.oxfamhosted.co.uk          60   IN CNAME [redacted].silentpush.[redacted]
[redacted].silentpush.[redacted] 300  IN A     49.[redacted]
```

Figure 8: Re-pointing www.oxfamhosted.co[.]uk to research domain.

```
dig www.oxjam.org

www.oxjam.org                 21600   IN    CNAME   www.oxfamhosted.co.uk
www.oxfamhosted.co.uk         60      IN    CNAME   [redacted].silentpush.[redacted]
[redacted].silentpush.[redacted] 300  IN    A       49.[redacted]
```

Figure 9: Confirmation of takeover of www.oxjam[.]org.

Step 4 – Configuring the new web server

Once the DNS had been fully repointed, we configured a webserver for `www.oxjam[.]org` with a basic HTTP site that receives traffic to the research webserver.

```
server {
       server_name www.oxjam.org;
       access_log /var/log/nginx/www.oxjam.org.access.log;
       error_log /var/log/nginx/www.oxjam.org.net.error.log;
       root /var/www/[redacted];
       index index.html;


    listen 443 ssl; # managed by Certbot•
    ssl_certificate /etc/letsencrypt/live/www.oxjam.org/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/www.oxjam.org/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

server {

    if ($host = www.oxjam.org) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

       listen 80;
       server_name www.oxjam.org;
    return 404; # managed by Certbot
}
```

Figure 10: Configuring the web server at 49.x.x.x.

Step 5 – Requesting an SSL certificate for www.oxjam[.]org

Once the web server had been provisioned, even though we didn't own `www.oxjam[.]org`, the initial dangling `CNAME` entry now led to all traffic to the domain being redirected to the research web server.

We then used *Let's Encrypt*, a free open certificate authority, to request an SSL certificate for `www.oxjam[.]org`. Even though the domain is owned by Oxfam, the certificate validation process is HTTP-based, hence the need to set up an HTTP site and get traffic flowing to validate the certificate request.

Step 6 – Upgrading the connection traffic to SSL and completing the takeover

Following the site setup, we ran a *Let's Encrypt* command-line tool that updated all the connection traffic to the web server to SSL, placing a validation icon in the visitor's browser that confirms `www.oxjam[.]org` as a legitimate domain with a valid certificate, despite it not being under the control of Oxfam.

Once this was completed, we obtained full control of `www.oxjam[.]org`, allowing the fake website to appear completely trustworthy, with a valid SSL certificate.

Step 7 – Confirming traffic flow and alerting the domain owner

Once full control had been obtained, we confirmed traffic flow to `www.oxjam[.]org`, spurred on by certificate transparency logs and indexes. The site was then configured to display a message warning Oxfam that their DNS configuration was open to exploit. Black hat hackers, however, would have had a multitude of exploits available to them, including embedded malware, fake payment portals requesting money for tickets, and other malicious redirects.

```
66.249.79.116 - - [21/Jul/2022:02:08:04 +0000] "GET / HTTP/1.1" 200 423 "-" "Mozilla/5.0 (Linux; Android 6.0.1; Nexus 5X Build/MMB29P) AppleWebKit/537.36
114.119.135.245 - - [21/Jul/2022:05:28:03 +0000] "GET /robots.txt HTTP/1.1" 200 32 "-" "Mozilla/5.0 (compatible;PetalBot;+https://webmaster.petalsearch.com/site/petalbot)"
123.6.49.36 - - [21/Jul/2022:08:22:42 +0000] "HEAD / HTTP/1.1" 200 0 "http://baidu.com" "Mozilla/5.0 (Linux; U; Android 8.0.0; zh-cn; Mi Note 2 Build/OPR1.170623.0
50.222.162.243 - - [21/Jul/2022:09:00:40 +0000] "GET / HTTP/1.1" 200 665 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
104.143.83.241 - - [21/Jul/2022:09:15:41 +0000] "GET / HTTP/1.1" 200 423 "http://www.oxjam.org/" "Mozilla/5.0 zgrab/0.x"
35.212.26.22 - - [21/Jul/2022:10:21:10 +0000] "GET / HTTP/1.1" 200 423 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko;
34.76.96.16 - - [21/Jul/2022:22:16:32 +0000] "GET / HTTP/1.1" 200 665 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
137.226.113.44 - - [21/Jul/2022:23:22:57 +0000] "GET / HTTP/1.1" 200 423 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:84.0) Gecko/20100101 Firefox/84.0"
50.222.162.243 - - [20/Jul/2022:08:51:47 +0000] "GET / HTTP/1.1" 200 665 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
104.143.83.241 - - [20/Jul/2022:09:13:35 +0000] "GET / HTTP/1.1" 200 423 "http://www.oxjam.org/" "Mozilla/5.0 zgrab/0.x"•
66.249.79.112 - - [20/Jul/2022:11:53:15 +0000] "GET /robots.txt HTTP/1.1" 200 32 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"
66.249.79.112 - - [20/Jul/2022:11:53:16 +0000] "GET / HTTP/1.1" 200 423 "-" "Mozilla/5.0 (Linux; Android 6.0.1; Nexus 5X Build/MMB29P) AppleWebKit/537.36 (KHTML, 104.192.108.9 - -
20/Jul/2022:15:48:56 +0000] "GET / HTTP/1.1" 200 423 "-" "Mozilla/5.0 (Linux; Android 8.0; Pixel 2 Build/OPD3.170816.012) AppleWebKit/537.36 (KHTM
171.13.14.46 - - [20/Jul/2022:15:48:59 +0000] "GET / HTTP/1.1" 200 423 "-" "Mozilla/5.0 (Linux; Android 8.0; Pixel 2 Build/OPD3.170816.012) AppleWebKit/537.36 (KHTML
50.222.162.243 - - [19/Jul/2022:08:54:52 +0000] "GET / HTTP/1.1" 200 665 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
104.143.83.241 - - [19/Jul/2022:09:15:33 +0000] "GET / HTTP/1.1" 200 423 "http://www.oxjam.org/" "Mozilla/5.0 zgrab/0.x"
128.14.209.154 - - [29/Jul/2022:01:53:50 +0000] "GET / HTTP/1.1" 200 423 "-" "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
137.226.113.44 - - [29/Jul/2022:02:14:54 +0000] "GET / HTTP/1.1" 200 423 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:84.0) Gecko/20100101 Firefox/84.0"
128.14.209.146 - - [29/Jul/2022:06:33:09 +0000] "GET / HTTP/1.1" 200 423 "http://www.oxjam.org/" "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML,
```

Figure 11: Traffic flowing to compromised site at www.oxjam[.]org.

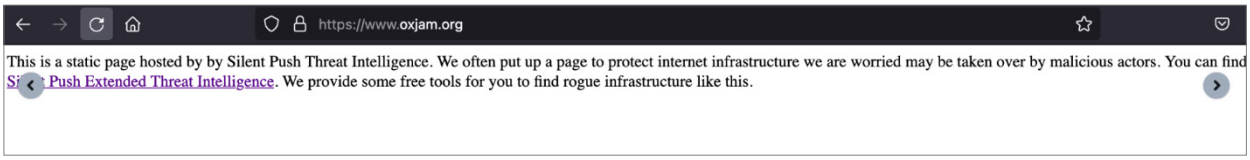

Figure 12: Holding page at www.oxjam[.]org.

Step 8 – Finding related domain targets via reverse lookup

Using the global DNS dataset, a *reverse* `CNAME` lookup was performed that scanned for targets across `*.oxfamhosted.co[.]uk` – a non-native DNS function that is only possible through DNS data collection.

Using this technique, we immediately identified various other Oxfam-related domains that now pointed to the research domain, as shown in Figure 13.

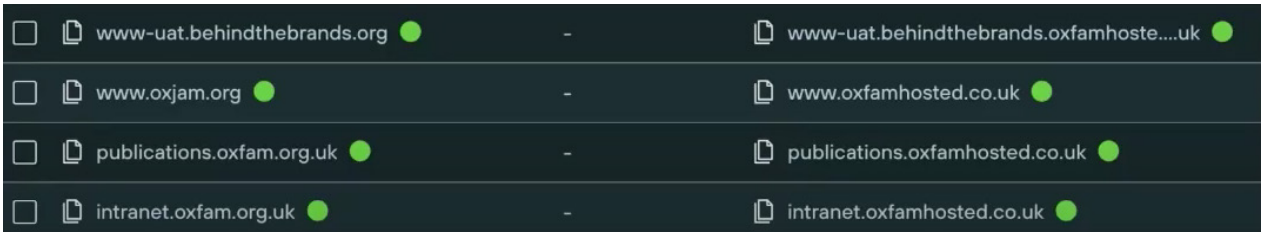

Figure 13: Related domains discovered via a reverse CNAME lookup.

Cookie harvesting

Although we ceased activity at this point, it is worth exploring what else could have happened, were it to have been a group of threat actors taking over Oxfam websites, and not a group of well-intentioned white hat researchers.

Cookie stealing is a particularly virulent kind of subdomain exploit that parlays the takeover into something far larger than traffic redirection through nameserver hijacking. Cookie harvesting works by threat actors configuring a page on attacker-controlled infrastructure, inside their control but also inside the subdomain scope of the target site.

From there, users are fooled into clicking on a masked link that points to the attacker's hostname but re-directs back to the initial site. Once the user's browser has successfully followed the masked link, it then deposits a copy of the user's domain cookies on attacker-controlled infrastructure, before re-directing back to the initial site. The redirection and harvesting are done with such speed, and masked in such a way, that the user would find it impossible to notice that they were visiting any site other than the one they were expecting to.

CONCLUSION

This study has shown that, given the number of records involved (see Table 3), dangling DNS records and subdomain takeovers are still not receiving the attention they deserve from the security community, relative to their risk level and their history within the global cybersecurity community. This is due to the inherent difficulty in large-scale DNS enumeration efforts that encompass the full scope of the internet's IPv4 range, coupled with the fragmented nature of the global DNS system and out-of-date administration practices.

Technology is being developed that offers organizations and security vendors the ability to quickly enumerate a list of DNS records related to a specific domain, from a pool of billions within an active DNS database solution that partitions records per type, to discover dangling entries. Private security vendors and brand owners need to migrate from brute force monitoring techniques that limit threat hunting to scripted queries within an incomplete (and therefore inadequate) pool of DNS records, towards all-encompassing scanning solutions that remove the financial and operational barriers to full and categorical DNS enumeration.

The study has also shown that poor DNS housekeeping – including the misconfiguration of `CNAME` and `NS` records, among others – contributes towards the proliferation of subdomain takeovers, which in turn leads to numerous other security exploits being propagated across the surface of a victim's domain scope, not limited to malicious content and extended to authentication exploits, among other open-ended outcomes that cause significant financial and reputational harm.

To safeguard against dangling DNS records and the open-ended peril of a subdomain takeover, organizations need to adopt robust DNS inventory management procedures that include regular daily monitoring of all DNS records, including (but not limited to) the auditing of zone files, proactive deletion of dangling records, auditing of connected third-party services, internationally recognized change management procedures that cater for DNS changes, clean-ups following infrastructure as code (IaC) activities, and awareness training among security teams.

The study calls on domain owners, security vendors and domain registrars to operate with a more robust set of inventory management, auditing, active monitoring, and change management protocols that improves DNS housekeeping and reduces the global supply of dangling records at source.

REFERENCES

[1] Petrova, D. DNS history. When and why was DNS created? ClouDNS. 30 May 2023. https://www.cloudns.net/blog/dns-history-creation-first/#:~:text=The%20DNS%20was%20created%20in,(Berkeley%20Internet%20Name%20Domain).

[2] Walfish, M.; Balakrishnan, H.; Shenker, S. Untangling the Web from DNS. NSDI. Vol. 4. 2004.

[3] Webber, J.; Parastatidis, S.; Robinson, I. REST in practice: Hypermedia and systems architecture. O'Reilly Media, Inc. 2010.

[4] Liu, D.; Hao, S.; Wang, H. All your DNS records point to us: Understanding the security threats of dangling DNS records. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. 2016.

[5] Help Net Security. Subdomain takeover attacks on the rise and harder to monitor. 29 March 2022. https://www.helpnetsecurity.com/2022/03/29/subdomain-takeovers-on-the-rise/.

[6] Rashid, S.M.Z.U.; Imtiaz Kamrul, M.D.; Islam, A. Understanding the security threats of esoteric subdomain takeover and prevention scheme. 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE). IEEE, 2019.

[7] Silent Push. We need to talk about subdomain takeovers. https://www.silentpush.com/blog/we-need-to-talk-about-subdomain-takeovers.

[8] Kintis, P. et al. Hiding in plain sight: A longitudinal study of combosquatting abuse. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017.

[9] Oxfam. Oxfam Annual Report 2021/22. https://www.oxfam.org.uk/documents/639/Oxfam_Annual_Report_and_Accounts_2021_22.pdf.