Zone Poisoning: The How and Where of Non-Secure DNS Dynamic Updates

Maciej Korczyński
Delft University of Technology
The Netherlands
Maciej.Korczynski@tudelft.nl

Michał Król
Université de Technologie de Compiègne
France
Michal.Krol@hds.utc.fr

Michel van Eeten
Delft University of Technology
The Netherlands
M.J.G.vanEeten@tudelft.nl

ABSTRACT

This paper illuminates the problem of non-secure DNS dynamic updates, which allow a miscreant to manipulate DNS entries in the zone files of authoritative name servers. We refer to this type of attack as to *zone poisoning*. This paper presents the first measurement study of the vulnerability. We analyze a random sample of 2.9 million domains and the Alexa top 1 million domains and find that at least 1,877 (0.065%) and 587 (0.062%) of domains are vulnerable, respectively. Among the vulnerable domains are governments, health care providers and banks, demonstrating that the threat impacts important services. Via this study and subsequent notifications to affected parties, we aim to improve the security of the DNS ecosystem.

Keywords

Domain Name System; zone poisoning; dynamic updates; measurement; security

1. INTRODUCTION

The Domain Name System (DNS) provides a critical service for all Internet applications that depend on domain names. Over the years, a variety of threats have emerged that undermine the trustworthy resolution of domain names into IP addresses. Two well-known attacks are cache poisoning [21] and malicious name resolution services [23, 16]. What these attacks share in common is that they compromise the resolution path somewhere between the user and the authoritative name server for a domain.

IMC 2016, November 14-16, 2016, Santa Monica, CA, USA
 ISBN 978-1-4503-4526-2/16/11…$15.00
DOI: http://dx.doi.org/10.1145/2987443.2987477

In this study, we explore an attack against the authoritative end of the path: the zone file of the authoritative name server itself. We detail how the vulnerable-by-design, non-secure DNS dynamic update protocol extension potentially allows anyone who can reach an authoritative name server to update the content of its zone file. The attacker only needs to know the name of the zone and the name server for that zone. The vulnerability was indicated already in 1997 by Vixie *et al.* in RFC 2136 [38], but its relevance in the current DNS landscape has not been recognized nor studied.

We refer to this type of attack as to *zone poisoning*. In the simplest version of an attack, a miscreant could replace an existing `A` or `MX` resource record (RR) in a zone file of an authoritative server and point the domain name to an IP address under control of an attacker.

We already know that criminals are interested in hacking DNS records of legitimate domains from the practice of *domain shadowing*, where registrant credentials are compromised in order to create a large volume of subdomains of a legitimate domain. They are used for, among other things, distributing malware exploit kits [13]. A more ambitious vector is hacking the registrars directly, as illustrated by the attack of Syrian Electronic Army on Melbourne IT, the registrar for the New York Times and Twitter [10]. In contrast to these attacks, zone poisoning does not require compromising registrants or registrars, but is as simple as sending a single RFC-compliant DNS dynamic update packet to a misconfigured server.

We present the first study to detail this vulnerability and measure its prevalence in the wild. Our main contributions are summarized as follows:

- We analyze the root cause of non-secure dynamic updates and how they can be exploited.
- We measure which domains allow non-secure dynamic updates in a random sample of 1% from 286 million domains and find that 0.065% is vulnerable. Surprisingly, we find a similar rate (0.062%, meaning 587 domains) for the Alexa top 1 million domains.
- Alarmingly, we find a significant number of domains

of national governments, universities, and businesses, including nine domains belonging to banks in Europe, Middle East, and Asia, from the domain of a private banking firm to a domain belonging to one of the largest banks in the world.

- We find significant concentrations of the vulnerability: securing the zone files of just 10 providers would reduce the prevalence of the issue with 88.6% in the random sample.
- We observe suspicious domains among the vulnerable population, but find no direct evidence of ongoing attacks.
- We find that most vulnerable servers are running Windows DNS, NLnetLabs NSD, and ISC BIND.

The objective of this paper is to strengthen the security of DNS. We notified all operators of non-secure servers discovered during our measurements.

2. BACKGROUND

The DNS protocol was initially designed to support queries of a statically configured database. Most of the data in the system was updated manually and expected to change only slowly [30]. However, with the introduction of dynamic allocation of network addresses to hosts [18], a more dynamic update mechanism for DNS became essential.

2.1 Dynamic Updates in DNS

DNS dynamic update specifications have been introduced by Vixie *et al.* in RFC 2136 [38] in 1997. Following this specification, one can add or delete any type of RR, such as `A`, `AAAA`, `CNAME`, or `NS`. The proposed `UPDATE` message complies with the standard DNS message format (cf. RFC 1035 [31]).

When a primary master server that supports dynamic updates receives an update request, it verifies: *i)* if all prerequisites defined by the requestor are met (e.g. check whether a specific record does or does not exist) and *ii)* whether restrictions are set regarding which hosts are allowed to make updates and, if so, whether those restrictions are met. If no restrictions are defined, anyone who knows the name of the zone and the name server for that zone is capable of updating its content. This constitutes a serious technological vulnerability indicated by Vixie *et al.* in RFC 2136 [38]. If the request is sent to an authoritative slave server, it is expected that it will be forwarded towards the primary server that is able to modify the zone file.

2.2 Secure DNS Dynamic Updates

Vixie *et al.* strongly recommended the use of security measures such as those described in RFC 2137 [14] (superseded by RFC 3007 [39]). If secure communication is not implemented, it is expected that an authoritative server accepts the dynamic updates only from a statically configured IP address of, for example, a DHCP server [38]. In RFC 2137, Donald Eastlake describes how to use the DNS Security Extensions (DNSSEC) [15] to restrict dynamic updates to authorized entities based on cryptographic keys [14]. However, using the public key mechanism is less efficient and harder to manage. Three years after the introduction of DNS dynamic updates, Vixie *et al.* proposed an efficient, lightweight alternative to authenticate dynamic updates: Secret Key Transaction Authentication for DNS (TSIG), which is based on shared secret keys and message authentication code (MAC) [34].

2.3 Implementations

We now analyze common implementations of DNS dynamic updates, paying special attention to the default protocol configurations.

BIND: Berkeley Internet Name Domain (BIND) is open source and the most widely used DNS software on the Internet [19]. Version 8, released in 1997, first included a dynamic DNS component [20, 37]. In BIND 8 and 9, dynamic updates are disabled by default. An administrator can add `allow-update` in the zone configuration and specify the hosts that are allowed to update records. An address match list can include entire subnetworks or the built-in argument `any`, that allows all hosts to make dynamic updates. Since BIND 8.2, released in 1999, the address match list supports TSIG. The basic configuration is still supported, however. Since BIND 9.1, slave servers are allowed to forward dynamic updates to a master server (RFC 2136 [9]). These can use address match lists similar to those of the master, meaning that non-secure configurations provide an additional path for a miscreant, as updates forwarded by the slave will be accepted by the master, regardless of the original requestor.

Microsoft DNS: Windows 2000 is the first operating system developed by Microsoft that supported DNS dynamic updates [28]. The server can be configured either as standard primary or as Microsoft's Active Directory–integrated zone [26]. Windows 2000 and its successors, i.e. Windows Server 2003 [25], 2008 [29], and 2012 [27], all support secure dynamic updates. They implement an extended TSIG algorithm (RFC 3645 [24]). When an administrator creates an Active Directory–integrated zone, by default the server allows only secure updates via extended TSIG. However, the server can also be configured for no or non-secure dynamic updates. More importantly, the secure update functionality is not available for standard primary zones. In any primary zone configured for DNS dynamic updates, anyone can modify zones.

Other Implementations: As indicated in RFC 2137 [14], any zone file allowing dynamic updates is less secure than the one configured statically. Some of the popular open-source authoritative servers such as Name Server Daemon (NSD) developed by NLnet Labs [32], DJBDNS created by Daniel J. Bernstein [12], or Unlogic Eagle DNS [33] do not support dynamic updates.

However, the functionality is sometimes added via external tools[1,2]. PowerDNS has recently added the dynamic update component. According to the documentation, by default all IP ranges are allowed to perform updates [35]. Our lab experiments (cf. Section 4.1) reveal, however, that by default only loopback IP space can make dynamic updates.

In short: common implementations not only support vulnerable configurations, such as accepting requests from all hosts, but some are vulnerable by default. Of the two common security mechanisms, TSIG-variants and address match lists, only the former provides a reliable defense to malicious updates. Since the attack only needs a single UDP packet, an attacker can guess and spoof source IP addresses on the match list. This risk could be mitigated by restricting dynamic updates to the TCP protocol only.

3. THREAT MODEL

We refer to an attack that exploits non-secure dynamic updates as *zone poisoning*. This attack itself is nothing more than sending a single RFC-compliant packet. The requirements are: *i)* non-secure updates are allowed by an authoritative server for a given zone *ii)* the miscreant knows the name of a zone and its name server.

An attacker can replace existing `A` or `MX` RRs in a zone file and point the domain to an IP address controlled by the attacker and potentially running a fake web or mail server. This would hijack the domain and allow the attacker to determine where clients or their emails go.

A miscreant could also abuse the reputation of a legitimate domain (e.g. `onlineshopping.com`) and add an extra `A` RR to an existing zone file that associates an IP address of a fake web server with a malicious subdomain (e.g. `paypal.account.onlineshopping.com`). An interesting variant is to *delegate* a malicious subdomain of a legitimate domain to the criminal's own DNS server. This would allow him to generate as many new subdomains as needed, without making additional update requests.

Non-secure updates could also be abused to acquire a Domain Validated (DV) SSL certificate for the vulnerable domain name, to be used in impersonation attacks. DV SSL certs are validated and provisioned automatically using a system of "challenge-response" emails. The attacker could re-route the confirmation message to the contact email listed in `WHOIS` via a dynamic update for the mail server domain.

4. METHODOLOGY

4.1 Lab Experiments

We performed lab experiments to establish if and how the protocol allows unauthorized dynamic updates, in particular adding, deleting and modifying existing records in the zone. We selected BIND 9.8.4 and PowerDNS 4.0.0-alpha2 as case studies, as both implementations are non-commercial and widely used. We configured master servers for our domain name (e.g., `example.com`) and we tested various configuration setups as explained in Section 2.3. To perform updates, we used both the standard Linux `nsupdate`[3] command and our own scanner (see Section 4.2). Updates were sent from both legitimate and spoofed source IP addresses on the address match list.

The update requests successfully added and deleted `A`, `AAAA`, `NS`, `MX`, `PTR`, `SOA` and `TXT` RRs corresponding to the domain name (`example.com`), as well as extra records for subdomain names (`researchdelft.example.com`). This way, we were also able to replace a pre-existing `A` RR (`example.com`) that had been manually added to the zone file at the beginning of the study. More specifically, using dynamic updates, we first added an extra `A` record that associated the domain name with a new IP address, and then removed the original one. Finally, for BIND we also configured the slave server to forward updates towards the master. As expected, the changes were accepted by the master even though the original requestor is allowed to make changes only in the slave server.

To conclude, our lab experiments demonstrate that systems which allow non-secure dynamic updates are vulnerable to attacks that can "modify" existing records and add new records. Non-secure update mechanisms cover both overly promiscuous address match lists ("`any`") as well as more focused match lists, which can be bypassed via IP spoofing.

4.2 Scanning Setup

To assess the potential impact of non-secure dynamic updates, we have developed an efficient scanner capable of sending DNS packets compliant with RFC 2136 [38]. The scanner attempts to add an extra `A` record to the zone file, associating a new upper-level domain, `researchdelft`, with the IP address of our project's web server. We do not spoof the source IP address of the update request. Our web server describes the project and provides a method to opt-out from our scans. Note that we have not received a single abuse complaint or opt-out request – which might mean that the insertion of the record was not seen as problematic or, perhaps more likely, that the insertion went unnoticed. The scan does not interact with the existing data in the zone file. Since our request is technically equivalent to a regular update request, we do not expect it to interfere with normal activity and have seen no evidence to the contrary.

We analyzed responses of authoritative name servers and performed DNS lookups to verify if our domain resolved to our web server's IP address. We also per-

[1]https://www.sixxs.net/wiki/NSD
[2]http://www.thismetalsky.org/projects/dhcp_dns
[3]http://linux.die.net/man/8/nsupdate

Table 1: Datasets

#	1% Sample	Alexa 1M
Domains	2,865,393	947,823
NS	510,850	487,515
IPs of NS	438,478	418,251
Domain–NS–IP	27,499,061	7,368,659

formed a ten-day long study to estimate the time the added RR stays in a zone. Finally, we removed the test DNS record by sending a *delete* `UDPATE` request and then tried to resolve it again. All added records were successfully deleted.

4.3 Ethical Considerations

While vulnerability scanning has become an established part of security research, our approach does raise ethical questions because of the fact that the only valid method available to us for assessing the vulnerability of a DNS server was to add a record to the zone file.

We have submitted the study to the TU Delft Human Research Ethics Committee. The committee evaluated our request and stated that we did not need their authorization since we were not conducting human subjects research. While this makes sense, it also signals that current institutional review procedures are not set up to evaluate ethical issues in computer security.

We have assessed our work using the principles outlined in the Menlo report [17]. We do not collect data on persons. Getting informed consent before adding a record to the zone file is both unpractical and would introduce selection bias, since administrators of well-secured servers are more likely to consent. We do provide a clear opt-out mechanism via the website referenced in the added DNS record. The site also provides full transparency regarding the study and its objectives.

Our approach in testing the vulnerability has been designed to have as minimal impact as possible: we send a single RFC-compliant packet. We do not read, change or otherwise engage with any existing records. We feel the drawback of lacking consent from server operators is outweighed by the benefits of our measurement for those operators: to be made aware of a critical vulnerability in their DNS server. All notifications have been completed before the publication of this paper. The new record is highly unlikely to be discovered by accident and it is removed at the end of the study.

4.4 Dataset

To measure the prevalence of non-secure configurations, we collected data for two samples: a random sample of 1% of the domain space and the Alexa top 1 million domains (or Alexa 1M) [1].

First, we extracted all domains observed in two complementary datasets between Jan 2015 and Jan 2016: *i)* DNSDB that is a large passive DNS database fed by hundreds of sensors across the world, operated by Farsight Security [3], which generously provided access to us and *ii)* Project Sonar Data Repository obtained though `ANY` RR requests, made available by Rapid7 Labs [4].

From the total 286,788,250 unique domains in the set, we randomly sampled 1%. For that sample and for the Alexa 1M, we enumerated all observed combinations of name servers and their IP addresses in both datasets: over 27 and 7 million, respectively (cf. Table 1). The long period of observation and the fact that DNSDB contains many entries that are poisoned either maliciously [23, 16] or unintentionally [40], means we expected a lot of IP addresses on the list to be obsolete, but we wanted to find as many as possible.

We performed the vulnerability assessment against the random sample on Mar 30, 2016 and against the Alexa 1M on Apr 10, 2016. For each domain, we sent an `UPDATE` request directly to all IP addresses on the list. As expected, many did not respond. Next to obsolete NS information, this can also indicate network filtering and other policies at work. We received responses from 6.0 million (random sample) and 2.3 million (Alexa 1M) name servers (see Table 2).

Table 2: DNS responses to `UPDATE` requests

DNS	1% Sample		Alexa 1M	
Response	in #	in %	in #	in %
All	6,007,462	100	2,294,099	100
`REFUSED`	2,325,377	38.7	1,265,544	55.2
`FORMERR`	1,374,015	22.8	260,094	11.3
`NOTAUTH`	1,198,337	19.9	357,442	15.6
`NOTIMP`	727,734	12.1	357,592	15.6
`SOA`	237,175	3.9	18,241	0.8
`SQR`*	114,677	1.9	25,851	1.1
`NOERROR`	13,580	0.2	5,093	0.2
`SERVFAIL`	6,621	0.2	3,830	0.2
Other	9,946	0.2	412	0

* Standard Query Response

5. RESULTS

5.1 Prevalence of Vulnerable Resources

Table 2 summarizes the DNS status codes received in response packets related to the `UPDATE` requests. As expected, the great majority of requests fail to add RRs to the zone. The most common code is `REFUSED`, meaning that the server refuses to perform the operation for security or policy reasons. Around 12.1% and 15.6% of name servers signal `NOTIMP` meaning that they do not implement the protocol extension, whereas 22.8% and 11.3% of servers are not even able to parse and interpret the dynamic update request and signal `FORMERR`. Next, 19.9% and 15.6% of name servers signal that they are not authoritative for the zone. The main reason for DNS responses with the `NOTAUTH` error flag is the presence of obsolete NS information in our dataset as described in Section 4.4. Approximately 0.2% of servers signal `SERVFAIL` meaning that a hardware error or an out-of-memory condition might have taken place and

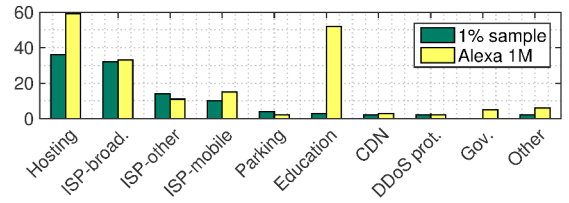


Figure 1: Types of providers hosting vulnerable domains.

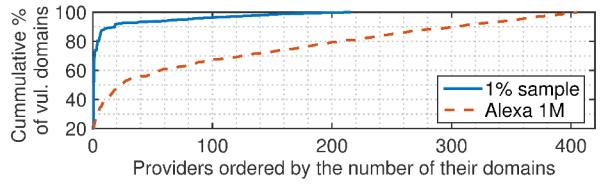


Figure 2: Cumulative distribution of vulnerable domains over providers.

a zone is restored to its state before this transaction [38]. We find 13,580 and 5,093 systems to respond with `NOERROR` status code for 1% sample and Alexa 1M respectively, which in both cases corresponds to 0.2% of responses. Note that `NOERROR` includes all responses with this status flag set regardless of whether the actual content of the zone has been updated.

We sent an `A` RR request to each of the potentially updated servers to verify if the zone file was indeed updated. For the random sample, we observed 2,626 successfully added `A` RRs, corresponding with 188 unique name servers and 1,877 unique domain names (0.065% of all randomly selected second-level domains). Surprisingly, we also observed 881 added `A` RRs that corresponded to 560 unique name servers and 587 domains from Alexa 1M (0.062%)

5.2 Affected Domains

To get a sense of the population of vulnerable domains, we first analyzed the type of network that hosts them. In earlier work, we developed a categorization of providers based on ground-truth data, manual labeling, `WHOIS` records and passive DNS data – for more details, see [11, 36]. We were able to classify 105 (out of 206) providers for the random sample and 210 (out of 398) for the Alexa 1M.

Figure 1 outlines the number of providers that have at least one vulnerable server in their network. As expected, hosting and ISP broadband constitute a great portion of the affected providers. Interestingly, we observe misconfigured zones in as many as 52 educational networks in the Alexa 1M.

Figure 2 shows the cumulative distribution of vulnerable domains over providers. In the random sample, we find that 66.2% (1,149) of vulnerable domains are hosted on the infrastructure of a single Japanese broadband ISP. Reconfiguring the zone files of just 10 providers would reduce the prevalence of the issue with 88.6%. If this kind of concentration is representative of

Table 3: Categories of vulnerable domains for Alexa 1M

Type	in #	in %
`Business`	181	31
`Entertainment`	92	15.7
`Educational`	90	15.3
`Governmental`	56	9.5
`News services`	41	7
`Adult`	13	2.2
`Financial services`	9	1.5
`Health care`	8	1.4
`Other`	95	16.2
Total	587	100

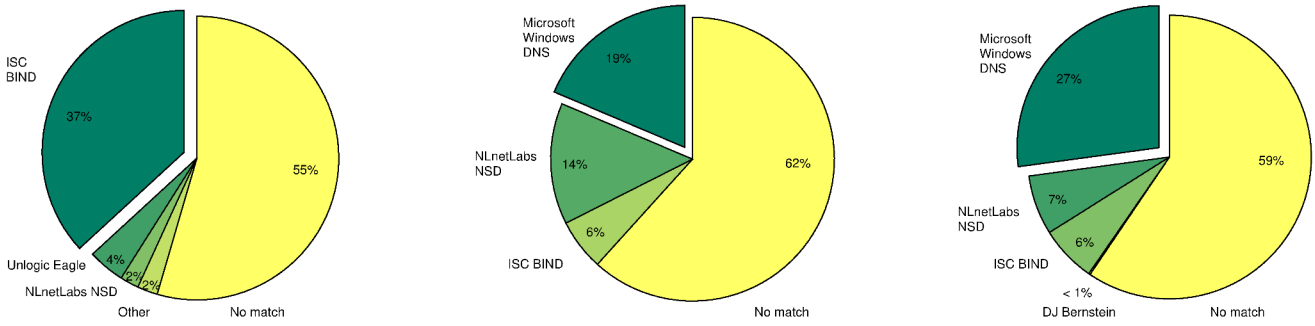
the overall domain space, then reaching out to a limited number of operators could greatly increasing the costs of finding vulnerable domains for cybercriminals. For the Alexa 1M, the pattern is much less concentrated. This might not be a major obstacle for remediation, though, as the high traffic sites in this set are typically professionally operated, so a comprehensive notification campaign might be effective.

We further analyze the cumulative distributions of vulnerable domains on DNS servers in descending order of the number of their common domains. For reasons of brevity, we highlight only the most interesting findings. In vulnerable 1% sample, we find that only one server is authoritative for as many as 1,635 (87%) domains, whereas in Alexa 1M, one DNS server is associated with 154 (26%) domains. As expected the cumulative concentrations per DNS servers are similar to the ones observed for providers (see Figure 2) as they operate the name servers themselves. In the 1% sample, for example, just six servers that share the same second- and top-level domain (`*.dnsserver.net`) are authoritative for 89.8% of the vulnerable domains, all hosted by the same broadband ISP in Japan.

We manually inspected the vulnerable domains from Alexa 1M. Table 3 lists the types of organizations affected. 'Business' is a large category that covers a heterogeneous set of companies, from small to large. In the latter category, we find a variety of sites related to global car manufacturers. We also find 56 vulnerable governmental sites in the North America, Europe, Asia – some national, some regional. Affected educational domains have a similar geographical distribution and include a few reputable universities. In health care, we found several hospitals and the domain of a national medical association. Remarkably, nine of the vulnerable domains belong to banks in Europe, Middle East and Asia, ranging from a small private banking firm to a domain of one of the largest banks in the world. In sum: the vulnerability is found to undermine the security of high-profile businesses, governments and organizations.

5.3 Exploitation

We looked for evidence of whether non-secure updates were exploited in the wild. We checked the overlap between the vulnerable domains and domains black-



(a) All 1% sample (b) Vulnerable 1% sample (c) Alexa 1M

Figure 3: Results for `FPDNS` fingerprinting of authoritative servers for (a) all 1% sample of the domain space, (b) vulnerable 1% sample, (c) vulnerable Alexa 1M domains (data may not sum up to 100% due to the round-off error).

listed by StopBadware [5] and the Anti-Phishing Working Group (APWG) [2] in 2015. The former consists of 1,016,961 unique fully qualified domain names (FQDNs) whereas the latter of 1,967,995. In APWG and StopBadware, respectively, we find 15 and 45 blacklisted FQDNs related to vulnerable second-level domains for Alexa 1M and only 1 and 5 for the random sample. After manual inspection of the website content, we did not find any compelling evidence that the observed domains are actually affected by malicious dynamic updates. The sites seemed legitimate and might either represent false positives or compromised resources.

We also searched in DNSDB for FQDN of vulnerable domains in association with common words in phishing attacks [8, 7], such as Paypal, Apple, Taobao, Amazon, etc. We find some suspicious FQDNs, for example, `shopping.*.com.*.*.edu` or `*.alibaba.com.*.ru`. However, the sites are either offline or require some additional authentication to access. Some of them seem legitimate proxy services, e.g., university resources that require authorized access and redirect users to an external website.

5.4 Affected DNS Server Software

We surveyed the software running on non-secure authoritative name servers to see which packages were affected. On Apr 24, 2016 we scanned three groups of servers by using `FPDNS` software [6]: *i)* all 510,850 name servers from the random sample, for comparative purposes; *ii)* the 188 vulnerable servers from the random sample; and *iii)* the 560 vulnerable servers from the Alexa 1M sample. Fingerprinting failed in many cases due to timeouts or inconclusive signatures. We were able to obtain software information for 45% (232,317), 38% (72), and 41% (227) of each respective group. We do not distinguish between different software versions as there are no major changes in the implementation of secure DNS dynamic updates (cf. Section 2.3). Figure 3 illustrates the results for DNS software fingerprinting. The majority of servers authoritative for the total random sample run BIND (37%). Microsoft Windows DNS constitutes just 0.5% of this group, while for the vulnerable groups it is the dominant package: 19% and 27%. The second and third largest groups of vulnerable server types are NLnetLabs NSD and ISC BIND. As the standard package of NLnetLabs NSD does not include the functionality for dynamic updates, we suspect that it might be added through some external, RFC-compliant plugin (see Section 2.3).

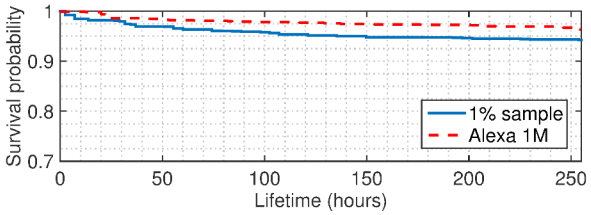


Figure 4: Survival analysis of `A` records added to vulnerable servers for 1% sample and Alexa 1M domains.

5.5 Survival Analysis

The final part of the study aimed to measure the survival times of the added records. We wanted to analyze whether these records would be removed and, if so, how soon. In other words, are there self-correcting mechanisms in place?

We initiated measurement on Apr 16, 2016. We first sent an update request to add an extra `A` RR (see Section 4.2) to the previously confirmed instances of vulnerable domains. We observe 3,920 successfully added `A` records that correspond to 1,870 domain names for 1% sample and 1,691 `A` RR associated with 584 domains for Alexa 1M domains.

Then, over a 10-day period, we performed DNS lookups every 4 hours—sending an `A` RR request to each of the IP addresses of the servers associated with vulnerable domains. We performed survival analysis on the results using the standard Kaplan-Meier estimator to approximate the survival function [22].

The results indicate a very small removal rate of the added record (cf. Figure 4). We do not know why some records were removed, but one plausible explanation is that the zone transfer from the primary master may

have overwritten the added entries. At the end of our experiment, records were still present in around 94.3% (3,696) of the random sample and 95.9% (1,622) of the Alexa 1M domains. Interestingly enough, the Alexa 1M does not have a higher removal rate than the random sample; in fact, it does slightly worse. In light of the fact that we were not contacted by any of the operators of the non-secure servers, suggesting no one saw the added record, it seems that there are no other security mechanisms in place to discover and mitigate the threat.

6. CONCLUSIONS

We presented the first measurement study into the vulnerability of non-secure DNS dynamic updates, which enables an attack we referred to as *zone poisoning*. We have measured prevalence rates for a random sample of 2.9 million domains (0.065%) and for the Alexa top 1 million domains (0.062%) and found that the vulnerability poses a serious security flaw that deserves more attention from domain owners and DNS service operators.

Certain limitations have to be taken into account to contextualize the obtained results. First, and perhaps foremost, we should note that our measurements establish a conservative lower bound for the magnitude of the problem. The servers that rely on address match lists to secure dynamic updates are counted as 'secure' in our measurement, but they are still vulnerable to IP spoofing. The attack requires only a single packet, making it possible for attackers to guess addresses that are on the match list.

The datasets in our study also present certain inherent limitations. For example, DNSDB has extensive, but not complete coverage of the domain name space. It also contains entries that are poisoned or obsolete, so many servers did not respond to our dynamic updates. Finally, we should note that responsibility is distributed and complicated. The fact that we found certain providers and software packages to be associated with vulnerable domains, should not be interpreted as assigning blame.

The next step for this work is to expand measurement and notify all affected parties, in order to improve the security of the DNS ecosystem, a critical service for many applications.

Acknowledgments

Authors thank Paul Vixie and Eric Ziegast from Farsight Security for sharing DNSDB, Jeroen van der Ham from the National Cyber Security Center (NCSC), Jelte Jansen, Moritz Müller and Marco Davids from SIDN, and the anonymous reviewers for their constructive and valuable comments. This work was supported by SIDN, the .NL Registry and by NWO (grant nr. 12.003/628 .001.003), NCSC. This work has been carried out in the framework of the project "IMATISSE" (Inundation Monitoring and Alarm Technology In a System of SystEms), funded by the Region Picardie, France, through the European Regional Development Fund (ERDF).

7. REFERENCES

[1] Alexa Top 1,000,000 Sites. http://s3.amazonaws.com/alexa-static/top-1m.csv.zip, Retrieved March 28, 2016.

[2] Anti-Phishing Working Group (APWG): Cross-industry Global Group Supporting Tackling the Phishing Menace.

[3] Farsight Security: DNS Database (DNS-DB). https://www.dnsdb.info.

[4] Internet-Wide Scan Data Repository: DNS Records (ANY). https://scans.io/study/sonar.fdns.

[5] StopBadware: A Nonprofit Anti-malware Organization. https://www.stopbadware.org.

[6] FPDNS-DNS Fingerprinting Tool. https://www.dns-oarc.net/tools/fpdns, 2014.

[7] Over a Quarter of Phishing Attacks in 2014 Targeted Users' Financial Data. http://www.kaspersky.com, February 2015.

[8] Aaron, G., and Rasmussen, R. Anti-Phishing Working Group (APWG) Global Phishing Survey: Trends and Domain Name Use in 2H2014. http://internetidentity.com/wp-content/uploads/2015/05/APWG_Global_Phishing_Report_2H_2014.pdf, May 2015.

[9] Albitz, P., and Liu, C. *DNS and BIND, 4th Edition*. O'Reilly Media, 2001.

[10] Arthur, C. Twitter and New York Times Still Patchy as Registrar Admits SEA Hack. https://www.theguardian.com, 2013.

[11] Asghari, H., van Eeten, M. J., and Bauer, J. M. Economics of Fighting Botnets: Lessons From a Decade of Mitigation. *IEEE Security & Privacy*, 5 (2015), 16–23.

[12] Bernstein, D. J. DJBDNS. https://cr.yp.to/djbdns.html, Retrieved March 2016.

[13] Biasini, N., and Esler, J. Threat Spotlight: Angler Lurking in the Domain Shadows. http://blogs.cisco.com, March 2015.

[14] D. Eastlake 3rd. Secure Domain Name System Dynamic Update. Internet RFC 2137, April 1997.

[15] D. Eastlake 3rd. Domain Name System Security Extensions. Internet RFC 2535, March 1999.

[16] Dagon, D., Provos, N., Lee, C. P., and Lee, W. Corrupted DNS Resolution Paths: The Rise of a Malicious Resolution Authority. In *Proc. of NDSS* (2008).

[17] Dittrich, D., and Kenneally, E. The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research. Tech. rep., U.S. Department of Homeland Security,

August 2012.

[18] Droms, R. Dynamic Host Configuration Protocol. Internet RFC 2131, March 1997.

[19] Internet Systems Consortium, Inc. BIND – The Most Widely Used Name Server Software. https://www.isc.org/downloads/bind, November 2015.

[20] Internet Systems Consortium, Inc. History of BIND. https://www.isc.org/history-of-bind, January 2015.

[21] Kaminsky, D. It's The End Of The Cache As We Know It. In: Black Hat Conference, http://www.slideshare.net/dakami/dmk-bo2-k8, August 2008.

[22] Kaplan, E. L., and Meier, P. Nonparametric Estimation from Incomplete Observations. *Journal of the American Statistical Association 53*, 282 (1958), 457–481.

[23] Kührer, M., Hupperich, T., Bushart, J., Rossow, C., and Holz, T. Going Wild: Large-Scale Classification of Open DNS Resolvers. In *Proc. of ACM IMC* (2015), pp. 355–368.

[24] Kwan, S., Garg, P., Gilroy, J., Esibov, L., Westhead, J., and Hall, R. Generic Security Service Algorithm for Secret Key Transaction Authentication for DNS (GSS-TSIG). Internet RFC 3645, October 2003.

[25] Microsoft TechNet. https://technet.microsoft.com/en-us/library/cc784052(v=ws.10).aspx, January.

[26] Microsoft TechNet. Active Directory-Integrated DNS Zones. https://technet.microsoft.com/en-us/library/cc731204(v=ws.10).aspx, April 2012.

[27] Microsoft TechNet. What's New in DNS Server. https://technet.microsoft.com/en-us/library/dn305898.aspx, June 2015.

[28] Microsoft TechNet. Dynamic Update and Secure Dynamic Update. https://technet.microsoft.com/en-us/library/cc959275.aspx, Retrieved March 2016.

[29] Microsoft TechNet. Understanding Dynamic Update. https://technet.microsoft.com/en-us/library/cc771255.aspx, Retrieved March 2016.

[30] Mockapetris, P. Domain Names - Concepts and Facilities. Internet RFC 1034, November 1987.

[31] Mockapetris, P. Domain Names - Implementation and Specification. Internet RFC 1035, November 1987.

[32] NLnet Labs. NSD: Name Server Daemon. http://www.nlnetlabs.nl/projects/nsd/, Retrieved March 2016.

[33] Olofsson, R. Eagle DNS. http://www.unlogic.se/projects/eagledns, Retrieved March 2016.

[34] P. Vixie, O. Gudmundsson, D. Eastlake 3rd, and B. Wellington. Secret Key Transaction Authentication for DNS (TSIG). Internet RFC 2845, May 2000.

[35] PowerDNS. Dynamic DNS Update (RFC2136). https://doc.powerdns.com/md/authoritative/dnsupdate, Retrieved March 2016.

[36] Tajalizadehkhoob, S., Korczyński, M., Noroozian, A., Gañán, C., and van Eeten, M. Apples, Oranges and Hosting Providers: Heterogeneity and Security in the Hosting Market. In *Proc. of IEEE NOMS* (2016), IEEE Press.

[37] Univerität Tübingen. BIND Version 8 Online Documentation. http://astro.uni-tuebingen.de/software/bind, March 1998.

[38] Vixie, P., Thomson, S., Rekhter, Y., and Bound, J. Dynamic Updates in the Domain Name System (DNS UPDATE). Internet RFC 2136, April 1997.

[39] Wellington, B. Secure Domain Name System (DNS) Dynamic Update. Internet RFC 3007, November 2000.

[40] Wessels, D. DNS Survey: Cache Poisoners. http://dns.measurement-factory.com/surveys/poisoners.html, 2007.