Saving Brian's Privacy: the Perils of Privacy Exposure through Reverse DNS

Olivier van der Toorn
University of Twente
o.i.vandertoorn@utwente.nl

Roland van Rijswijk-Deij
University of Twente
r.m.vanrijswijk@utwente.nl

Raffaele Sommese
University of Twente
r.sommese@utwente.nl

Anna Sperotto
University of Twente
a.sperotto@utwente.nl

Mattijs Jonker
University of Twente
m.jonker@utwente.nl

ABSTRACT

Given the importance of privacy, many Internet protocols are nowadays designed with privacy in mind (e.g., using TLS for *confidentiality*). Foreseeing all privacy issues at the time of protocol design is, however, challenging and may become near impossible when interaction out of protocol bounds occurs. One demonstrably not well understood interaction occurs when DHCP exchanges are accompanied by automated changes to the global DNS (e.g., to dynamically add hostnames for allocated IP addresses). As we will substantiate, this is a privacy risk: one may be able to infer device presence and network dynamics from virtually *anywhere* on the Internet — and even identify and track individuals — even if other mechanisms to limit tracking by outsiders (e.g., blocking pings) are in place.

We present a first of its kind study into this risk. We identify networks that expose client identifiers in reverse DNS records and study the relation between the presence of clients and said records. Our results show a strong link: in 9 out of 10 cases, records linger for at most an hour, for a selection of academic, enterprise and ISP networks alike. We also demonstrate how client patterns and network dynamics can be learned, by tracking devices owned by persons named *Brian* over time, revealing shifts in work patterns caused by COVID-19 related work-from-home measures, and by determining a good time to stage a heist.

CCS CONCEPTS

• **Networks → Naming and addressing**; **Network management**; **Network privacy and anonymity**; Network dynamics.

KEYWORDS

Reverse DNS, DHCP, Privacy, Tracking

ACM Reference Format:
Olivier van der Toorn, Roland van Rijswijk-Deij, Raffaele Sommese, Anna Sperotto, and Mattijs Jonker. 2022. Saving Brian's Privacy: the Perils of Privacy Exposure through Reverse DNS. In *ACM Internet Measurement Conference (IMC '22), October 25–27, 2022, Nice, France.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3517745.3561424

IMC '22, October 25–27, 2022, Nice, France
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9259-4/22/10.
https://doi.org/10.1145/3517745.3561424

1 INTRODUCTION

Privacy violations frequently headline the news. Notorious incidents from the past years have often involved threat actors, questionable data processing practices, or human error. Given the importance of privacy, many Internet protocols are nowadays designed with it in mind (e.g., using TLS for *confidentiality*). Still, it is challenging to foresee all privacy issues at protocol design, and this may be infeasible if interaction out of protocol bounds can occur.

One protocol that has prompted privacy concerns is the Dynamic Host Configuration Protocol (DHCP), which is a network management protocol that is used to dynamically assign IP addresses to devices on a network. DHCP uses a client-server model and allows for client devices to send optional communication parameters to the server. A number of research efforts have focused on DHCP privacy, demonstrating that locally monitoring and sniffing DHCP messages – which can contain unique client identifiers – enables geotemporal tracking of clients, even if clients move between networks [2, 3, 25]. Groat et al. [10] even introduce the idea of remotely monitoring DHCP client devices, using compromised DHCPv6 relays inside the monitored network.

DHCP exchanges can be accompanied by changes to the Domain Name System (DNS). DHCP servers can update *local* name services, to associate devices on the local network with a name. Servers can also make changes to the *global* DNS, for example by adding hostnames (i.e., Pointer (`PTR`) records) for allocated IP addresses. `PTR` records can then be publicly accessed using Reverse DNS (rDNS) lookups, mapping IP addresses to hostnames. RFC 7844 [12] recognizes the privacy risk of carrying over unique client identifiers from DHCP options to DNS, but the extent to which this happens in practice has received little attention in the research literature. Furthermore, making automated changes to DNS records based on DHCP exchanges is in itself a privacy risk that seems to have stayed under the radar so far. As we will demonstrate, this practice allows network dynamics to be observed from virtually *anywhere* on the Internet. The risks resulting from this are that one can track the presence of client devices in networks, may be able to reliably identify specific devices and tie these to persons, and might even be able to track clients across multiple networks.

In this paper, we perform a first of its kind study into these risks. Our goal is to substantiate that the interaction between DHCP and DNS leads to unwanted leaks of potentially privacy-sensitive information, which is then disclosed to the public Internet due to the open nature of the DNS. We demonstrate the existence of this threat by investigating rDNS data. Our results show that there is a disparity between the guidance in standards with respect to privacy and DHCP and DNS interaction, versus the actual implementation in practice. This disparity is what allows for sensitive information to leak to the public Internet.

The contributions of this paper are:

(1) We show that DNS records contain unique identifiers in practice, even including sensitive information such as client device types and device owner names.
(2) We demonstrate that networks of varying types (academic, enterprise, ISP) expose such information.
(3) We analyze the relation between the presence of dynamically added hostnames in the DNS and the presence of client devices on networks.
(4) We demonstrate that outsiders can use reverse DNS to track specific clients and learn network dynamics.
(5) We discuss possible causes and ways to mitigate risk.

The remainder of this paper is organized as follows. In Section 2 we provide background information and discuss related work. We introduce our data sets in Section 3. We detail how we identify networks that expose dynamic behavior Section 4. In Section 5 we focus on leaking privacy-sensitive information in rDNS records. We look at timing of dynamically added rDNS entries in Section 6. Then, in Section 7, we present a number of case studies to show how client patterns and network dynamics can be learned. In Section 8 we discuss our findings and possible steps toward mitigating the privacy risk. Finally, we document ethical considerations in Section 9 and conclude in Section 10.

2 BACKGROUND & RELATED WORK

2.1 Background

In this section we provide background information on *reverse* and *forward* DNS, the DHCP, and IP Address Management (IPAM) systems.

DNS. The DNS is a critical component of the Internet and can be seen as its *phonebook*. It is responsible for translating between human-readable names and IP addresses. The DNS is operated as a distributed hierarchical database, in which parts of the namespace are delegated to different parties [18]. The DNS is typically used to translate domain names to IP addresses, which is referred to as *forward* DNS. The DNS also enables *reverse* resolution, in which an IP address is translated to a hostname. The DNS uses special zones for the purpose of the latter. The `in-addr.arpa.` zone is used to translate IPv4 addresses to hostnames [19]. This zone contains so-called `PTR` (Pointer) records, which one can query for by using the reversed form of the IP address one wishes to translate. With the reversed form, the DNS can be queried similarly as with forward DNS, except that a `PTR` record is requested, rather than an `A` record. Example 1 provides an example of an IP address and its reversed form for a `PTR` query.

What?	Value
IP address to translate	`93.184.216.34`
Reversed DNS query	`34.216.184.93.in-addr.arpa.`

Example 1: Reverse IPv4 Example

DHCP. The Dynamic Host Configuration Protocol (DHCP) is a network management protocol that can be used to dynamically assign IP addresses to devices on a network, using a client-server model. When a client device wishes to join the network using DHCP, it issues an address request to DHCP servers, either via broadcast or unicast in the event a server address is already known. The DHCP server can offer and subsequently allocate an IP address to the client for a set amount of time. Upon allocation, the client is told which address it is allowed to use and for how long. This allocation is called the DHCP lease and before it expires, the client can request renewal. If a lease expires, the associated IP address is considered reallocable. When a client leaves the network it can signal to the DHCP server, through a so-called release message, that it is in the process of leaving the network. The server can then reallocate the IP address sooner. Release messages are not always sent, as clients can go out of range, or users can unplug devices from the network abruptly.

DHCP client-server exchanges involve more information than the allocated IP address and expiration time. The server can convey communication parameters such as the default gateway. The client in turn can send information such as an optional `Host Name` [1] or `Client FQDN` [23]. The prior is commonly used by DHCP servers to identify hosts and also to update the address of the host in *local* name services. The latter would allow the server to update *global* name services, if the client so desires (cf. §3.3 in [23]).

IPAM Systems. IP Address Management (IPAM) systems allow network operators to manage various parts of IP address infrastructure. These systems are typically used in larger enterprises, where manually managing IP address space (e.g., assigning subnets or IP addresses) is no longer feasible. IPAM systems can be used to manage DHCP as well DNS.

Interplay between DHCP and DNS. DHCP and DNS can be linked together, through IPAM systems or by other means. If they are linked, then when a client requests a DHCP lease and is allocated an IP address, various changes to the DNS related to the IP address are made automatically. For example, the DHCP server can update information for the allocated IP address in local name services, which the server may do on the basis of the previously mentioned, client-provided `Host Name`.[1]

Automated changes to DNS are not limited to *local* name services per se. Changes to the *global* DNS can also be made. For example, upon allocation of an IP address, a hostname can be associated with said address by adding a `PTR` record to the global DNS. Evidently, if changes to the (public) DNS are made as client devices join or leave a network, one may be able to infer network dynamics by capturing DNS changes. The privacy risk further increases if client-provided parameters are used in DNS updates, as these may allow for specific devices to be identified and their presence or absence to be tracked. RFC 7844 [12] recognizes the potential risk of carrying-over unique identifiers to the DNS. As we will show, the makes, models and even owner names of devices running the DHCP client can appear in DNS data (e.g., *Brian's iPhone*).

[1] Note that `hostname` is commonly used to refer to the name to which an IP address translates (i.e., the name in its PTR record). To avoid ambiguity, we use `Host Name` to refer to the DHCP client parameter.

2.2 Related Work

We consider several types of related work: gaining information from hostnames in rDNS, DNS and privacy, and DHCP and privacy.

Information in hostnames. It is known in the literature that reverse DNS can reveal information about hosts. Various works exist in which authors successfully use hostnames to gain insights into Internet infrastructure and topology. Central to such works is the notion that hostnames can encode meaningful information. Chabarek et al. [6] used rDNS data to study part of the Internet core, by inferring link speeds of router and switch interfaces from hostnames. Huffaker et al. instead extract geographic information, using a dictionary of city names and airport codes [11]. Follow-up works by Luckie et al. focus on learning how to extract autonomous systems and network names from hostnames [16, 17].

The aforementioned works focus mostly on core infrastructure such as routers. Lee et al. [15] instead shift focus to end-users, i.e., customers of Internet Service Providers (ISPs). In their paper, they study means to infer the connection types of hosts in access networks. Zhang et al. [28], in turn, infer and geolocate topology in regional access networks, with the aim of studying architectural choices made by ISPs.

While all these works extract meaningful information from hostnames, to the best of our knowledge no works exist that consider what can be learned about networks by observing *automated* and *continual* changes to rDNS records. We bridge this gap and reveal that rDNS changes can provide insight into network dynamics and client behaviors. A common assumption in related work also seems to be that meaningful information is included in rDNS by network operators on purpose. We instead substantiate that reverse DNS can inadvertently contain privacy-sensitive information.

DNS and Privacy. The DNS is no stranger when it comes to privacy considerations on the Internet. In fact, DNS privacy has received a lot of attention over the past years. Often, the focus is on keeping interactions with the DNS confidential, or on aspects of DNS data sharing and processing [13, 14, 26, 27]. An example is QNAME-minimization [5, 7], which helps improve privacy by minimizing information sent from the recursive to upstream name servers. Other examples involve efforts to add confidentiality by encrypting queries and answers, for example via DNS-over-HTTPS (DoH) or -over-TLS (DoT). Generally speaking, studies that relate to the DNS often involve *forward* DNS. The *reverse* side of DNS is less frequently studied. Tatang et al. [24] studied privacy leaks in rDNS to a certain extent, after observing that some misconfigured name servers provide outsiders with answers to PTR queries for private IP addresses (e.g., RFC 1918 [9]), if such addresses are used inside the networks of the misconfigured servers. In their paper they characterized the country and network distribution of such servers, and studied privacy-sensitive patterns in hostnames, revealing end-user devices as well as security-critical infrastructure such as firewalls.

DHCP and Privacy. The DHCP protocol has given rise to privacy concerns, which led to discussions in the Dynamic Host Configuration (DHC) working group. These discussions resulted in RFC 7844 [12], which recognizes that DHCP client messages can contain unique client identifiers. Such identifiers can be used to track clients, even if devices take care of randomizing other link-layer identifiers such as MAC addresses. RFC 7844 also recognizes that identifiers can carry-over to the DNS. They propose anonymity profiles, which minimize disclosure of client-identifying information in DHCP messages.

Tront et al. [25] proposed using a dynamic DHCP unique identifier (DUID) based on the same randomization technique used in IPv6 privacy extensions. Groat et al. [10] show how the use of DHCPv6 to overcome the privacy issue of SLAAC deployment can still lead to the possibility to track users because of the use of static DUIDs. They also note that remote tracking may be possible, via compromised DHCPv6 relays that forward messages to attackers. Bernardos et al. [3] showed how randomization of L2 addresses was a convenient solution to mitigate location privacy issues on public Wi-Fi connections, evaluating also user experience and potential IP address pool exhaustion. Aura et al. [2] investigated how DHCP can be used to provide mobile clients with authenticated network location information, which clients can then use to decide how to behave in specific networks. In their paper, they consider the privacy of mobile users, by minimizing client information in DHCP messages, at least until the network has been authenticated. Finally, while the previously discussed work by Tatang et al. [24] did find patterns in the DNS that likely resulted from DHCP carry-over, the authors appear to not have considered the role of DHCP.

The literature has thus established that DHCP messages create opportunities to track client devices, even between subnets and networks. Central to most of these works is the ability to observe messages, which requires observer presence in network. Our work instead focuses on the risk associated with the interplay between DHCP and DNS. We extend and substantiate the risk that theoretically existed and study the problem in the wild.

3 DATA SETS

We use three data sets in this paper: two large-scale data sets of IPv4 rDNS measurements, and a smaller data set with ICMP and rDNS measurements that we collect ourselves.

Full address-space reverse DNS measurements. The bulk of our analysis relies on measurement data from reverse DNS measurements that cover the entire IPv4 address space. Several projects make rDNS data sets available for research. In this study, we use data sets collected by Rapid7's Project Sonar [22] and by the OpenINTEL project [21]. The Rapid7 data set is collected on a single weekday every week and OpenINTEL collects daily snapshots. Given that our goal is to show evidence of dynamic address assignments relating to human behavior, we prefer the data from OpenINTEL because of its daily measurement frequency. Where we need data that predates the first collection date obtainable from OpenINTEL, we use data from Rapid7 instead. Jointly, both data sets cover the period of our study: 2019-10-01 to 2021-12-31. Table 1 shows summary statistics on the data used from Rapid7 and OpenINTEL. The table details the number of data points in each data set as well as the daily average number of unique PTR records observed by each measurement.

Reactive fine-grained measurement. The third data set that we use in this work is a custom, supplemental measurement, described in Section 6.1. This data set includes both data from a dedicated

Table 1: Statistics for the data sets that we obtained from Rapid 7 and OpenINTEL.

	Start date	End date	Total # responses	# unique PTRs
Rapid7 Sonar	2019-10-01	2021-01-01	77 G	1,381 M
OpenINTEL	2020-02-17	2021-12-01	396 G	1,356 M

ICMP measurement and fine-grained data from a reactive rDNS measurement. The supplemental measurement was performed from 2021-10-27 to 2021-11-16. Summary statistics for this supplementary data set are given in Table 3.

4 IDENTIFYING DYNAMICITY EXPOSURE

In this section, we discuss how we identify networks that expose dynamic behavior through rDNS entries. We then proceed to apply our identifying methodology to our data set.

4.1 Methodology

To study whether networks dynamically add and expose privacy-sensitive information through rDNS, we first need to identify which networks exhibit signs of dynamic behavior in our data sets. We focus only on such networks because we aim at investigating temporal patterns of client devices. Other networks can still expose privacy-sensitive information in PTR records (i.e., hostnames) in a non-dynamic sense. However, these networks cannot be leveraged to learn temporal patterns related to clients, therefore we do not include them in our analysis. To identify dynamic networks, we use the daily data sets obtained from OpenINTEL and apply a set of heuristics in three steps detailed below.

Step 1: First, we perform a daily analysis over data covering a three-month period. We group results by /24 prefix and compute the unique number of IP addresses for which we see a PTR record on each day. We then discard the /24 prefixes for which we never observe more than 10 addresses a day over the three-month period under consideration. For the prefixes for which we do, we also record the maximum number of daily IP addresses per /24 over the three-month period.

Step 2: Next, we perform a day-by-day comparison for each /24 considered in over the three-month period and record the absolute difference in number of IP addresses for which we observe a PTR record. We then compute the "change percentage" by dividing this difference by the maximum number of addresses recorded in the previous step.

Step 3: Finally, we label /24 prefixes as *dynamic* if the change percentage exceeds $X\%$ on at least Y days over the entire three-month period. We set X to 10 (which sets the threshold at a single address changed for blocks with 10 or more addresses), and Y to 7, based on experiments.

Validation: We validate our heuristic approach against our own campus network. The addresses for this network come from a single /16 prefix with a numbering plan in which some subprefixes are used for dynamic allocations whereas other subprefixes contain static allocations. We run our approach to identify change-sensitive /24 blocks. Our method marks 40 prefixes as dynamic, and 206 prefixes as static. We confirmed these results with our campus ICT department. The 40 prefixes we identify as dynamic are confirmed as true positives. In addition to this, our IT department indicated a further 83 prefixes use dynamic address assignments (DHCP), but with static rDNS entries (i.e., fixed-form PTR records such as host1234.dynamic.institute.edu). This confirms that our heuristic approach correctly identifies prefixes with dynamically updated PTR records.

Threshold and dynamicity: our dynamicity methodology strongly depends on setting thresholds (for the change percentage X and the number of days Y). Given the values that we chose, we discard a large number of /24. Our rationale behind choosing such strict thresholds is that we want to identify dynamic networks with high confidence. As the preceding validation involving ground truth demonstrates, our threshold choices are reasonable. Thus, using these thresholds we establish a lower bound of dynamicity exposing networks.

4.2 Identifying Networks

We start out by identifying networks that exhibit dynamic behavior using the approach detailed in Section 4.1. We use the three-month period from 2021-01 to 2021-03 to identify such networks. Over this period, we see PTR records for a total of 6,151,219 unique /24 networks. Out of these, 134,451 are marked as dynamic using our heuristic approach. This result demonstrates that there is alarming evidence of networks exposing dynamics in (global) rDNS.

To gain a further intuition on how dynamic behavior in rDNS is visible as part of a larger network, we map any /24 prefix that we identify as dynamic back to the most-specific announced, covering prefix. Figure 1 shows the distribution of the fraction of /24 prefixes that make up a prefix that exhibit dynamic behavior. As the plot shows, generally speaking, only a small subset of the prefixes that make up a network exhibit dynamic behavior. An intuition for this result is the use of numbering plans, where specific subprefixes are used for dynamic clients (recall how this is done in our own campus network from the validation of our heuristic approach in Section 4.1). We leave further study of external visibility of such network segmentation as future work. Finally, we note that the result in Figure 1 also guides the choice of which subprefixes to subject to our supplemental measurement, which we return to later in Section 6.1.

5 IDENTIFYING PRIVACY LEAKS IN RECORDS

In this section, we identify the publication of privacy-sensitive information in rDNS and demonstrate an associated risk.

5.1 Methodology

Recall that our goal is to identify privacy-sensitive information in dynamically updated rDNS entries. In order to zoom in on such privacy leaks, we perform further filtering of our data sets, consisting of the following steps:

Extracting Common Terms. We start by analyzing terms that commonly appear in rDNS records. To find terms we use a regular expression that extracts words consisting of alphabetical characters from PTR records, of which we can count occurrences.

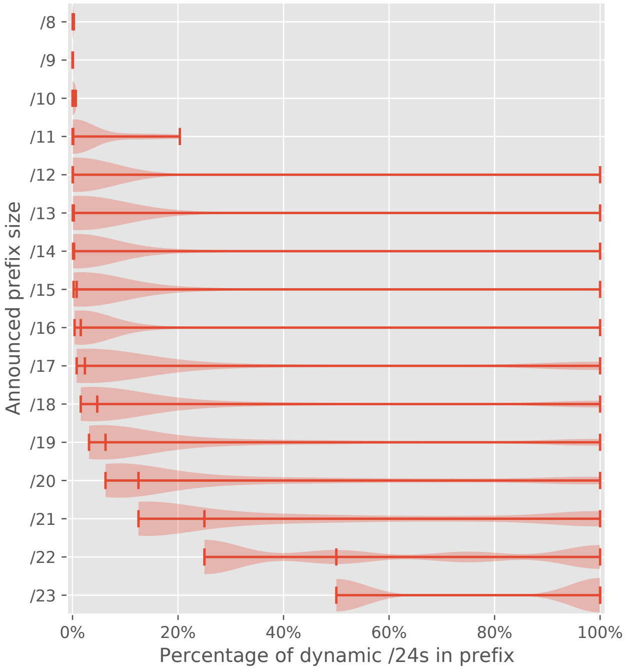


Figure 1: Distribution of the fraction of /24 prefixes that show dynamic rDNS behavior as part of the most-specific announced prefix they are part of. Ticks show the minimum, median and maximum number of /24 subprefixes that show dynamic behavior.

Common Suffixes. Hostnames for related IP addresses can have a common hostname suffix, to which host-specific parts are prepended. Consider, e.g., `client1.someisp.com` and `client2.someisp.com`. We identify suffix keywords (`someisp` and `com` in this example).

Generic Terms. Among non-suffix keywords, we identify a number of generic terms that convey location or router-level information. These terms are less likely to be used in client hostname prefixes. Examples are `north` and `south`. We use these terms to exclude router-level PTR records (see also [16, 17]).

Given Names. From the remaining PTR records we then select those that contain *given names*, as given names can be indicative of a user client device hostname. The US government keeps track of and publishes names given to newborns.[2] We select names for the years 2000 up to 2020, ranked by popularity over this 20-year period. We select the top 50 most popular names.

Dealing with City Names. Router-level hostnames can encode location information such as city names [11], which can overlap with *given names* (e.g., *Jackson* and *Jacksonville*). Instead of excluding such mismatches via enumeration (e.g., using a list of city names), we count the number of unique *given name* matches per hostname suffix and require this to be above a certain threshold. Our reasoning here is that if dynamic client devices are present, the number of uniquely matched *given names* will be relatively larger than the number of unique *city names* in router-level hostnames.

5.1.1 Identified Networks. We apply the aforementioned steps to daily rDNS data from OpenINTEL to find networks that likely add rDNS records for dynamic client devices and carry over unique identifiers to the DNS. Henceforth we refer to the found networks as the *identified networks*.

1. We start from the set of networks showing dynamic behavior, based on the heuristic approach described in Section 4.1.
2. We then exclude rDNS entries with generic router-level terms.
3. We match the remaining PTR records against a list of given names.
4. We extract hostname suffixes from the results and calculate per suffix: (1) the number of records; (2) the number of uniquely matched given names; and (3) the ratio between the two.
5. We select suffixes with at least 50 unique given name matches.[3]
6. We further require a ratio of 0.1 or more[3], to find a variety of matched names in sizeable networks.

5.2 Signs and Causes of Privacy Trouble

We now zoom in using the additional filtering steps described in Section 5.1. After filtering, we identify 197 networks that meet our strict criteria. We index these networks by hostname suffix (TLD+1) and manually classify them by network type. We start by making a number of general observations about these networks.

Given names. Figure 2 shows the number of given name matches in the rDNS data. The blue bars account for any matching PTR record. The red bars only count records that belong to networks that meet the uncertainty-minimizing thresholds and criteria set out in Sections 4.1 and 5.1. Figure 2 shows that given names are generally more common in prefixes that show dynamic behavior. Please note that, due to the logarithmic y-axis, the difference between the number of matches before and after filtering easily amounts to an order of magnitude.

Common Appearances in Hostnames. We investigate which words commonly co-appear alongside *given names* in hostnames, postulating that these terms may enable insight into how given names ended up in hostnames to begin with. From the common terms that occur a hundred times or more we manually select those that we think reveal information about client devices other than the user's *given name*.

Figure 3 shows the terms we selected, along with the number of PTR records in which the terms appear, before and after imposing the strict thresholds. The frequent co-appearance of terms such as `iphone`, `android` and `galaxy` are a strong indication that DHCP clients on a variety of mobile devices send the name of the device to the DHCP server, seeing as phone names can be formed of the owner's name and make or model (e.g., *Brian's iPhone*). The appearance of terms such as `laptop` and `desktop` are indicative of behavior of DHCP clients on other types of devices.

[2] https://www.ssa.gov/oact/babynames/

[3] We note that our aim is not to exhaustively identify and study all dynamic networks, but rather to identify a small set which likely leak privacy-sensitive information through rDNS entries. See also Section 9 in which we discuss ethical considerations.

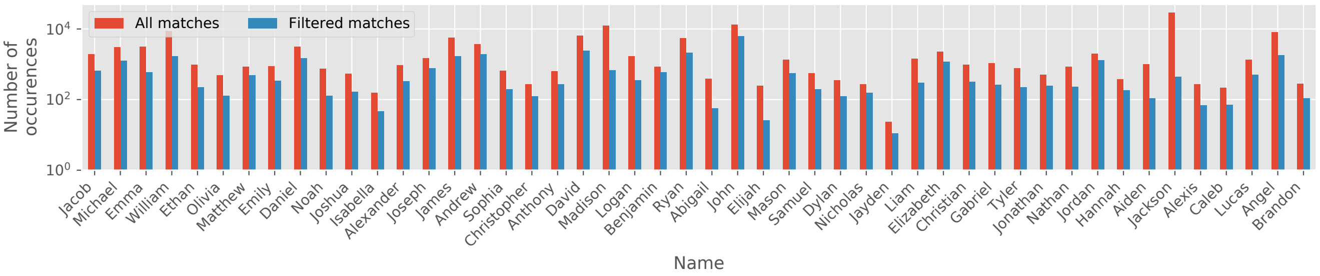


Figure 2: Given names for newborns (Top 50 sorted by US popularity) as observed in reverse DNS entries. The plot shows the total number of matches and the number of matches after filtering the networks (logarithmic scale).

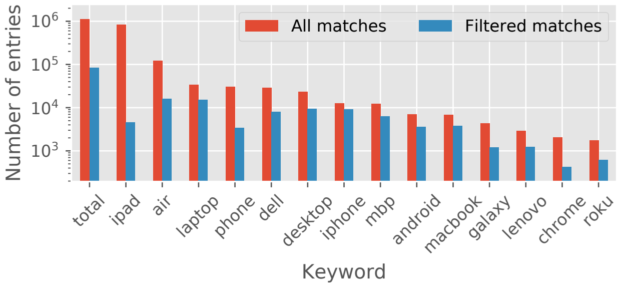


Figure 3: Terms frequently in hostnames along given names, before and after imposing *given name* thresholds (logarithmic scale). The column *total* is the sum, not a term.

As previously discussed in Section 2, RFC 7844 recognizes the risk of carrying over unique client identifiers from DHCP to DNS because these identifiers can be used to track clients [12]. Our findings do not only demonstrate that identifiers are in fact carried over in the wild, but also reveal that the content contained in identifiers is in itself privacy-sensitive. For example, being able to tell the make and model of a client device may benefit sophisticated attackers, who could use this information to pre-select relevant exploits. Owner names, in turn, can tie IP addresses to users, which could be used for a number of malicious purposes.

We suspect that phone and computer names are sent via the DHCP `Host Name`. While we do not claim that DHCP clients are necessarily at fault here, we do note that these terms can help identify the makes and models of devices and that this may require mitigation steps.

Beyond Given Names. While the approach we chose to identify networks hinges on the appearance of *given names*, we recognise that some commonly co-appearing terms can also be used independently. As our aim is not to exhaustively identify networks, this is not something we explore in this paper. We note that we considered terms of three or more characters. While shorter terms do co-appear, they add a lot of noise. As an example, consider the term hp, which may indicate HP laptops and desktops, but can also be a substring in other terms.

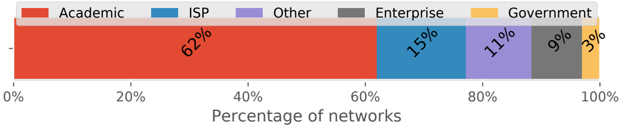


Figure 4: Breakdown of the 197 networks over the types *academic, ISP, enterprise, government* and *other*.

Network Types. We use a manual selection process to infer the type of each identified network by looking at hostname suffixes. The specific types that we identify are: *academic*, *ISP*, *enterprise*, *government* and *other*. We use regular expressions to match records against `.edu` and `.ac`, both of which indicate *academic* use, as well as `.gov` for *government* use. To find other network types such as *ISP* and *enterprise* networks, we use manual inspection. Figure 4 shows a breakdown of the results for the 197 *identified networks*. The majority of networks, 61.9%, is *academic* – these contain school, university and research institution networks. *ISPs* account for 15.2% of the networks, 9% are *enterprise* networks and 3% are classified as *government* networks. Finally, 11.2% have the label *other*, indicating that we were unable to clearly classify their type.

Key takeaways: *There are strong signs that networks of various types expose the presence of dynamic clients in rDNS. A problem beyond merely carrying over unique client identifiers from DHCP messages to the DNS becomes apparent: the makes, models and even owner names of client devices can be learned.*

6 TIMING OF RDNS ENTRIES

This section first explains our supplemental measurement to collect finer-grained timing information on dynamic client behavior in networks that exhibit dynamicity in rDNS entries. Next, we present our findings on the timing of rDNS records.

6.1 Methodology

The highest temporal measurement granularity of the two existing rDNS data sets that we use is daily. This means we cannot infer sub day level dynamicity (e.g., devices joining and leaving the network). To perform a more detailed study of the timing of rDNS entries and attempt to capture network dynamics beyond what can be learned

Table 2: Reactive measurement and back off strategy.

Number of measurements and measurement intervals
12 times in the 1st hour at 5-minute intervals
↪ 6 times in the 2nd hour at 10-minute intervals
↪ 3 times in the 3rd hour at 20-minute intervals
↪ 2 times in the 4th hour at 30-minute intervals
↪until client goes offline once at 60-minute intervals

from daily rDNS measurement data, we perform a supplemental measurement against the IP address space of a subset of *identified networks*.

Network Selection. To comply with the requirements from our IRB (see Section 9), we select a minimal subset from the 197 *identified networks* for supplemental measurement and further validation. We select nine networks – three networks of the three most-common types *academic*, *ISP* and *enterprise* (from Section 5.2). In the selection of three *academic* networks we favor one particular network as we have *a posteriori* knowledge about IP address distribution that has utility towards one of our case studies (Section 7.3).

We order the list of networks of each type by the number of *given name* matches and start selecting from the top. We perform an additional, manual inspection of PTR records to ensure that the networks we select show evidence of dynamically assigned hosts. We make a weighted selection of which address space of selected networks to target with supplemental measurement. For large networks, we dig a little deeper to observe which IP subnet (/16 or more specific) contains the most dynamically assigned hosts, and target this address space only. Whether or not networks respond to ICMP ping scans does not factor into the selection process.

Measurement Mechanics. Our supplemental measurement technique to investigate the timing of rDNS records involves two types of probing: (1) ICMP probes; and (2) finer-grained reverse DNS lookups. We run an hourly ICMP ping scan against the selected networks to determine if client devices have joined or left the network since the previous hour, provided of course that the devices respond to pings.

We hypothesize that client devices on a network go through the following three phases:

(1) The client joins the network and is allocated an IP address. An rDNS entry is added or updated by the DHCP server. The client device may start to respond to ping requests.
(2) The client is active on the network. In this phase it keeps responding to pings and the PTR remains unchanged.
(3) The client leaves the network and no longer responds to pings. The address may be deallocated and the PTR may be changed or removed. This is subject to behavior of the DHCP server and may also depend on whether or not the client releases its lease (see Section 2.1).

Phases 1 and 3 can speak to the relation between the presence of a client and the presence of an rDNS record. To measure this, we trigger reactive measurements when we infer, from the hourly ICMP ping scan, that a client has newly appeared on the network. We perform a reactive ping and trigger an rDNS query. We then continue with pings at five-minute intervals and gradually back off. After pinging 12×, which takes one hour, we reduce to ten-minute intervals during the second hour. The full back off schema is shown in Table 2. Once we infer that the client is no longer reachable, we reactively perform rDNS lookups for the IP address in question, following the same back off schema. Figure 5 shows a graphic representation of our supplemental measurement.

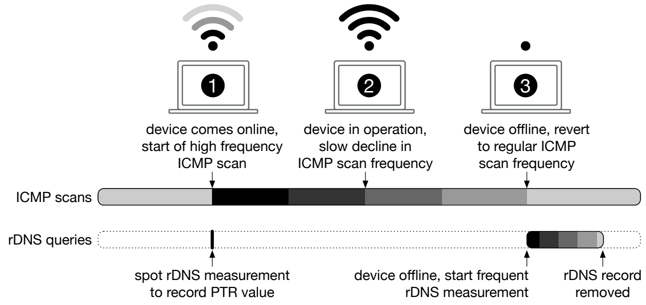

1
device comes online, start of high frequency ICMP scan
2
device in operation, slow decline in ICMP scan frequency
3
device offline, revert to regular ICMP scan frequency
ICMP scans
rDNS queries
spot rDNS measurement to record PTR value
device offline, start frequent rDNS measurement
rDNS record removed

Figure 5: Graphic representation of the mechanics of the supplemental measurement. Time flows from left to right, the two bars represent when ICMP and rDNS measurements are scheduled, numbers correspond to device activity phases.

We use Zmap [8] for the ICMP measurements. Zmap allows us to easily implement rate limiting and IP address blocklisting. The blocklisting capability is used to allow subjects to opt-out (see Section 9). We have set up our measurement infrastructure such that information about the measurement is easily findable, and contact details are given to allow opt-out of the measurement.

For the rDNS measurement we use custom-built software wrapping *dnspython*.[4] We rate-limit requests to authoritative name servers to reduce the impact of our measurement on the DNS name servers as much as possible. We query the authoritative name server for the IP address in question directly, to make sure we get a fresh answer (i.e., not from a cache). Both Zmap and our custom-built software write the results as CSV files to disk. Zmap only includes hosts that were reachable in its output.

Supplementary Measurement Data. The supplementary rDNS data may contain resolution errors, which come in the form of NXDOMAIN, authoritative name servers failing to answer, or timeouts. In our data set it is clear which are correct PTR responses and which are errors. The shortest follow-up time is five minutes. For this reason we add, next to the original timestamp, a truncated timestamp per five minutes to the ICMP and rDNS measurement data points. We can then merge supplementary measurement data based on IP address and timestamp. We next determine the start and endpoints of client activity by relating measurement data points. We are mostly interested in knowing what happens to the rDNS after a client joins or leaves. However, by considering rDNS data from around the time of the client joining, we can also verify if rDNS state is reverted after a client has left.

[4] https://www.dnspython.org/

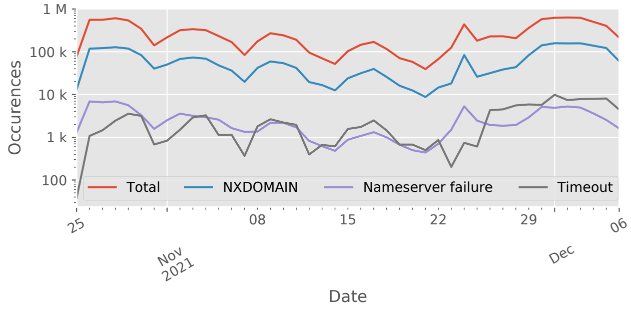


Figure 6: DNS errors observed during the supplemental measurement.

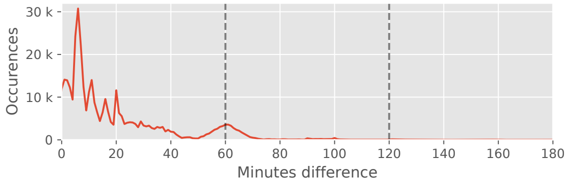
We assign activity at the IP address level and give each address, start and end point combination a group ID. This group ID allows supplementary measurement data to be tied to specific client activity periods. As a last step we aggregate by group ID, including the timestamps of the last ICMP and rDNS measurements, and the first and last measured rDNS entries. For groups to be usable towards making inferences, each should include at least successful ICMP probes and rDNS lookups for phases 1 and 3 (the client joining and leaving the network). If the group's data shows that the rDNS record is added and then removed, we can reliably infer a relation between the rDNS record and observed client activity, which allows us to investigate the (temporal) relation between the presence of clients on the network and the presence of rDNS records.

6.2 Observations of Client Activity

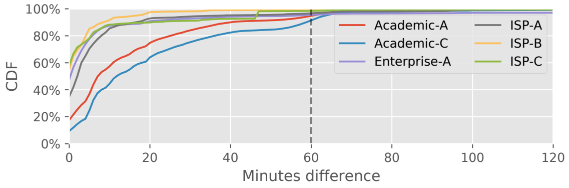
Table 3 shows summary statistics on the supplemental measurement, which we ran from 2021-10-27 to 2021-12-05. Table 4 shows additional, per-network information related to the supplemental measurement. The nine (anonymous) networks are shown, along with their type, size of the targeted IP address space, and number of addresses that respond to ICMP probes. For two out of three *enterprise* networks, we do not see responses to ICMP pings at all. We suspect the operators of these networks block pings on ingress. For one *academic* network (Academic-B) only two hosts responded to ICMP pings consistently throughout the measurement, but these particular IP addresses do not have PTR records. While we receive responses from within all three *ISP* networks, the responsiveness rates vary. We suspect that, in these networks, no blocks are imposed by the operator, and responsiveness thus fully depends on hosts being online.

We reiterate that the networks were selected because they show strong signs of adding dynamically assigned hostnames to rDNS. So even for the networks (or hosts) that block ICMP probes, rDNS data *can* be used to learn client device presence. In addition, rDNS queries reveal the hostname attached to the IP address, something ICMP probes do not provide.

Our supplemental measurement resulted in errors at times. Next to the normal responses, we saw name server failures, timeouts, and NXDOMAIN responses. Figure 6 shows the number of errors compared to the number of IP addresses seen per day (note the logarithmic y-axis). Fortunately, the number of errors is low relatively to the number of queries performed. In traditional DNS sense, receiving an NXDOMAIN response is seen as an error. In our case, however, this is a bit more nuanced. Depending on the time frame, a PTR could be missing because it is yet to be added to the DNS (phase 1 of client activity) or already removed (phase 3).[5]



(a) Absolute numbers of groups for a given (x-axis) minutes difference. First three hours are shown.



(b) CDF of the differences per network shown for the first two hours.

Figure 7: Difference in minutes between last ICMP sample and rDNS sample per measurement group.

As explained in our methodology (Section 6.1), we group supplementary measurement data points. Table 5 shows a breakdown of the groups in the supplementary data, starting with all measurement groups, down to those that can be used to make reliable inferences. The first group subcategory, *successful responses*, ensures the group has successful ICMP probes and rDNS lookups for the client joining and leaving the network (i.e., no timeouts or errors). The next subcategory, PTR *reverted*, involves groups in which we observe that the PTR is changed and reverted during and after the client's inferred presence. Of this subcategory, in about 1 out of 4 cases, timing mechanics of the ICMP probes, which cannot be accounted for at run-time without compromising the back off mechanism, make the results less reliable. After filtering these out, we are left with 419,453 usable groups.

Validating Timing Aspects. We can now shed light on the temporal relation between the presence of clients on the network and the presence of rDNS records. Figure 7a shows the number of groups offset against the time between a client leaving the network and PTR removal. The peaks around multiples of an hour suggest that PTR records are removed due to the expiration of a DHCP lease, which is often set to an hour for a fast turn-over rate. The peak close to the five minutes mark can be explained by clients cleanly leaving a network (i.e., by sending the optional DHCP release message upon leaving the network). If the DHCP server or IPAM system removes the PTR, we would see this five minutes later as the (respective) probes are sent in five-minute intervals.

[5] Sending additional PTR lookups for phase 1 would result in fewer inconclusive measurements, but the phase 3 issue cannot be corrected for during measurement.

Table 3: Supplemental measurement statistics.

	Start date	End date	Total # responses	# of unique IP addresses observed	# of unique PTR records observed
ICMP	2021-10-25	2021-12-05	45,496,201	80,738	-
rDNS	2021-10-25	2021-12-05	11,731,348	54,456	180,614

Table 4: The 9 networks targeted for supplemental measurement, along with their type, the size of the targeted address space, and number of addresses that respond to ICMP probes.

Network name	Network size	Addresses observed	Percent observed
Academic-A	/16	31,454	48.0%
Academic-B	/16	2	0.0%
Academic-C	/16	21,602	33.0%
Enterprise-A	/17, /19	24,055	58.7%
Enterprise-B	3 * /16	0	0.0%
Enterprise-C	5 * /24	0	0.0%
ISP-A	3 * /22	1,073	34.9%
ISP-B	/16, /17, /18	357	0.3%
ISP-C	/16	1,102	1.7%

Table 5: Breakdown of supplemental measurement results, down from all groups to those enabling inferences.

	#groups	Fraction of parent
All groups	6,297,080	100.0%
Successful responses	582,814	9.3%
PTR reverted	581,923	99.9%
Reliable timing alignment	419,453	72.1%

Figure 7b breaks down timing information for the individual networks targeted for supplemental measurement. The networks *Enterprise-B* and *Enterprise-C* are not shown, because no ICMP ping responses were received from these networks. *Academic-B* is not shown because the two hosts responding to ICMP did not have a corresponding rDNS entry. The CDFs show that in about 9 of 10 cases, the rDNS entries reverted within 60 minutes of a client leaving the network. We already demonstrated that the presence of client devices on the network can be learned from rDNS, which anyone can query. The fact that records do not linger long in most cases escalates the privacy risk: the timing enables observation of network dynamics. The differences in lingering between *Academic-A* and *Academic-B*, as apparent in Figure 7b, can be explained by a longer DHCP lease time. If these networks predominantly update rDNS in response to leases expiring, rather than to DHCP release messages, the rDNS records linger longer.

Key takeaways: *In networks that expose the presence of dynamic clients, there is a strong link between the existence of a* PTR *record for an IP address and the presence of a client device, which has been assigned that address. In 9 out of 10 cases,* PTR *records linger for 60 minutes or less after client disappearance.*

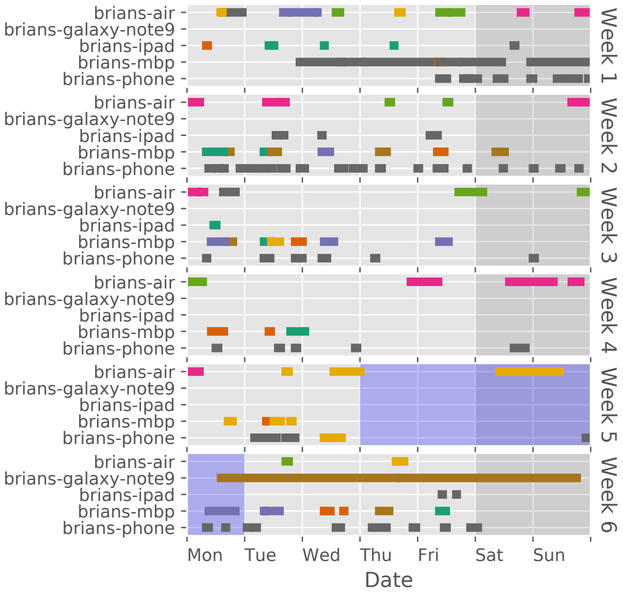


Figure 8: Six weeks in the Life of Brian(s). Weekends (gray) and Thanksgiving weekend (purple) are highlighted through shading. IP addresses are encoded with colored bars.

7 CASE STUDIES

Now that we have demonstrated that rDNS entries can be used to infer the presence of client devices on networks, we present a number of case studies. These case studies serve to look at some of the consequences and further substantiate the privacy risk.

7.1 Life of Brian(s)

To demonstrate the severity of the privacy risk, we use rDNS data to follow persons named *Brian* over time. For this we assume that the *given name* in the hostname reflects the name of a network client's owner. We use our supplementary rDNS measurement data for this case study. It is important to note that while we reactively looked up PTR records with ICMP pings as the trigger point during supplemental measurement, anyone with the capability to do frequent PTR lookups can capture the same patterns that we discuss in this case study (i.e., no ICMP required).

We use the data of the network *Academic-A*, which is an academic network in the US with campus housing. Figure 8 shows six weeks of client hostnames containing the *given name* Brian on *Academic-A*. We have color-coded IP addresses in the figure. Times in the figure are in the local timezone. Our intuition is that these hostnames are not related to a single *Brian*, but rather two or maybe three. The use of a private and work phone is not uncommon, but multiple laptops in active use arguably is. The clients `brians-air`, `brians-mbp` and `brians-phone` show regular patterns. Especially `brians-mbp` in week two shows very regular activity: a couple of hours around noon, every day.

We chose *Academic-A* and these six weeks because of the Thanksgiving weekend (the weekend of the fifth week). Thanksgiving is a US holiday in which many students go home to be with their families. Thanksgiving is always on a Thursday. In 2021, it fell on the 25th of November. The Friday and Monday after Thanksgiving are known as *Black Friday* and *Cyber Monday*, and many stores promote sales around these days, typically on electronics. In our results, `brians-phone` and `brians-mpb` seem to leave the network around Thursday. Striking is that `brians-galaxy-note9` appears in the afternoon on Cyber Monday. We have not observed this hostname before this time. We speculate that a Brian may have bought a Samsung Galaxy Note 9 during the Black Friday or Cyber Monday sales.

This case study shows that rDNS data provides insights into the behavior of clients to which hostnames are dynamically assigned. Since the hostnames contain given names, this may even be tied to specific individuals. If one knows or were able to infer how addresses are assigned to, e.g., specific buildings [28] on campus, one could track, from virtually anywhere on the Internet, a *Brian* around campus as he goes from lecture to lecture.

7.2 Working from Home

For the second case study, we move away from tracking specific (*Brian*-owned) clients over time and focus on investigating the overall dynamics of select networks instead. We use the OpenINTEL-provided rDNS data for this case study, to demonstrate that daily PTR measurements already offer insight into network dynamics. We select all three *academic* networks for this case study, as well as *enterprises* B and C, as these five networks show a specific change in user behavior that we wish to highlight in this case study.

For each of the selected networks, we calculate the total number of PTR records on any given day (i.e., we do not require given names to be present, etc.). Our aim is to see if there is a correlation between the presence of rDNS entries and lockdown regulations due to the COVID-19 pandemic. We expect enterprise networks to experience a drop in daily entries as employees were required to work from home. For academic networks this may be a bit more nuanced. Education buildings may see fewer clients, but with on-campus student housing, the client concentration may shift without it necessarily being visible in the total number of rDNS entries.

Figure 9 shows for each network the percentage of rDNS entries relative to the maximum number observed, over 2020 and 2021. The three academic networks are shown at the top. The selected enterprise networks at the bottom. We compare the public COVID-19 related news reports of *Academic-A* (shown in red in the figure) with the presence of rDNS entries. Upon making this comparison, we see a clear connection between the two.[6] For the times at which a moderate or high risk was reported to students and staff, sharp decreases in daily rDNS entries records are visible. After reports of a low risk of COVID-19 prevalence on campus, a sharp increase in network client device presence is visible.

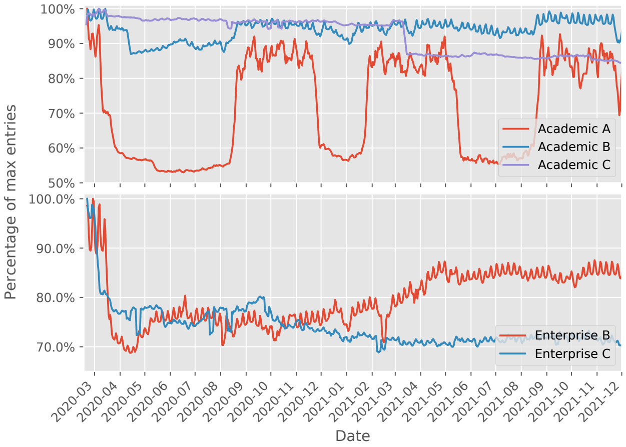


Figure 9: A longitudinal breakdown of reverse DNS entry presence for select networks, starting near the beginning of the COVID-19 outbreak. The percentage shown are of the number of entries relative to the maximum observed number of entries in each network.

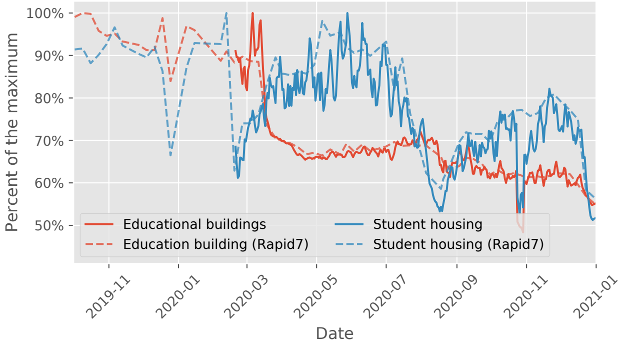


Figure 10: Zoomed in version of `Academic-C`, starting late 2019. Dashed lines are based on Rapid7 data. Solid lines are based on OpenINTEL data. Rapid7 provides weekly snapshots of rDNS state. OpenINTEL started daily rDNS measurements in February 2020.

For *Academic-B* we observe a marked reduction in rDNS entries during the first period of COVID-19 lockdowns, after which the number goes back up to about 95% of what we observed before the start of the pandemic. By September 2021, the level returns to that of before the pandemic, with the dip at the end corresponding to the Christmas holiday break.

The networks of *Enterprise-B* and *Enterprise-C* show significant decreases in rDNS entries in March and April of 2021. It stands to reason that these decreases are related to COVID-19 measures. *Enterprise-B* shows a partial recovery in the number of entries around May of 2021, which could be a sign of loosened restrictions, either by the government or by the employer.

[6] We do not link to these reports to protect the identity of network *Academic-A*.

In Figure 10 we look at network *Academic-C* in more detail. As this is the home institution of the authors, we know which IP subnets are used for on-campus student housing and educational buildings, and when buildings were closed. In this graph we show both Rapid7 and OpenINTEL data. The COVID-19 lockdown measures were introduced shortly after OpenINTEL started its rDNS measurements (2020-02-17). We use Rapid7 data, which has weekly granularity, to extend visibility into the early months of 2020.

In March a crossover between PTR records for educational buildings and student housing is clearly visible, signifying that: employees are working from home, educational buildings are empty, and students study from their on campus residences. The decreases at the end of October for both the educational buildings and student housing corresponds with the fall holiday week. A similar drop is visible at the end of the year for the 2020 Christmas break.

In absolute numbers (not in figure), the reverses for the educational buildings remain much higher than for student housing, which can be explained by having more address space assigned to educational buildings, with more static hosts online. The Rapid7 curves in Figure 10 (dashed) largely overlay and confirm the observations we make from OpenINTEL data. Additionally, given that Rapid7 data extends visibility into 2019, we can see that the number of network clients before the crossover is relatively stable. The 2019 Christmas break is also visible in Rapid7 data, as well as a drop towards the end of February 2020 that likely relates to Carnaval celebrations.[7]

While our case study into compliance with work-from-home measures is relatively harmless, it does show the extent to which even rDNS measurement data of daily granularity can be used to learn network dynamics, possibly for nefarious purposes. Our approach to observing work-from-home patterns using rDNS data adds adds to existing efforts in the literature towards this end (e.g., observing shifts in traffic volumes).

7.3 When to stage a heist?

Suppose that you want to stage a heist. There is something valuable in a building and you want to steal it while the least amount of people are around. Ideally, from the robber's perspective, they are able to determine the point in time at which the fewest dynamic clients are around. This evidently requires high-frequency rDNS measurements. For ethical reasons, we have not instrumented such a measurement. For this case study, we rely on data from our supplemental measurement instead.

We consider one week of supplementary data for the network *Academic-A*. This network responds to ICMP pings, which we use to support our findings. We stress, again, that ICMP responsiveness is not required. Even networks that block ICMP may be observed via rDNS, as our second case study shows (recall from Section 6.2 that networks *Enterprise-B* and *Enterprise-C* do not respond to ICMP probes).

Figure 11 shows the number of active clients inferred in the network *Academic-A* between 2021-11-01 and 2021-11-07, both for rDNS lookups (blue) and ICMP probes (red). The diurnal pattern of the network is visible, with most activity during the day and into the evening, while the least activity is at night and early in the morning. The rDNS measurements (blue) give a rough indication of the best time for the heist. As an example, on weekdays the data hint at approximately 6AM as a good time. We also show activity based on ICMP responses (red) for comparison and to support our findings. The ICMP results for the most agree. For networks that do not block ICMP, the robber could of course also use ICMP probes. In absolute numbers, the rDNS lookups pan out lower than the number of ICMP probes, which is due to the reactive nature of the rDNS measurement.

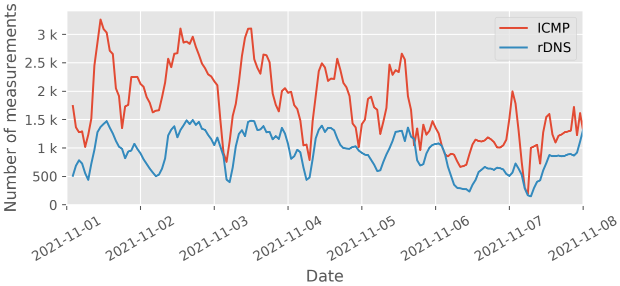


Figure 11: One week of measurements from *Academic-A* to demonstrate when one might stage a heist.

This case study shows the feasibility of one example of how outside observations of dynamically assigned hostnames can be used for nefarious purposes. These observations can help a potential attacker to learn working patterns without being physically present at the location. Our supplemental measurement is reactive and does not try to establish the number of clients on a network at any given time. A targeted measurement at a higher frequency would likely give better results. We leave a study to confirm this as future work, as this is out of scope for this paper.

8 DISCUSSION

Our findings are disconcerting. While existing literature has shown that meaningful information can be extracted from hostnames primarily without considering continual changes to reverse DNS records, we reveal that observing automated changes to rDNS can provide insights into client presence and network dynamics. The publicness of rDNS severely increases this risk, enabling anyone on the Internet to observe automated changes. An adversary with measurement capability and knowledge about a potential target can gain valuable insights following an approach similar to ours. We keep the invasiveness of our case studies in check and tailored our approach, but given recent findings that hostnames can encode building locations [28], it appears feasible that for some networks, rDNS data can be used to geotemporally track users at the building level.

An arguably sensible mechanism to limit the tracking of network client devices by outsiders is blocking ICMP pings at network ingress. Two of the networks we used for validation do not respond to ICMP pings. At the same time, the records these networks dynamically add to the global DNS, as well as the time these records linger after clients have left, allow anyone who frequently queries for rDNS records to observe the presence of clients in these networks.

[7] A local Catholic holiday.

A notion that other works that study hostnames have in common is that meaningful information is encoded in hostnames on purpose, especially for router-level entries. Our results substantiate that the interplay between DHCP and DNS can inadvertently provide anyone with DNS lookup capability insights into end-user client identifiers. Our results also reveal a more severe problem: privacy-sensitive information such as device owner names appear in the *global* DNS. While our validation covers nine networks, this problem is likely not limited to these networks. Our results thus confirm a risk outlined in RFC 7844 [12]: DHCP clients send revealing information in optional parameters. Based on our observations of terms commonly co-appearing with *given names* (e.g., `brians-ipad` and `brians-galaxy-note9`), we suspect that client implementations on various makes and models of phones and computers send *device names* to the DHCP server. While our choice to match against popular names given to US newborns creates bias towards these names, we accept this bias as we set out to substantiate the privacy risk rather than to exhaustively identify all rDNS records that contain privacy leaks.

Steps towards Mitigating the Problem. After raising these issues we would like to start the discussion on how to solve the problem. Evidently, the interplay between DHCP and DNS and the extent to which configuration and protocols permit client identifiers to flow from one protocol to another is at the core of this problem. While we have not investigated this extensively, we identified a number of IPAM softwares that make it easy to automate DNS changes. For example: Bluecat,[8] Efficient IP,[9] Infoblox,[10] Men & Mice,[11] and Solarwinds.[12] It is unclear to us if and which DHCP servers or IPAM systems come with default settings that carry over client identifiers to the global DNS. We would argue that it is rarely a good idea to indiscriminately carry over DHCP client-provided information such as *device names* to publicly accessible `PTR` records. Using some sort of hash seems prudent, although this may make hostnames less sensible. While we have not thoroughly investigated reasons for device manufacturers to send *device names* to DHCP servers, we know that for Bluetooth and Wi-Fi Direct pairing, sharing such information helps identify the device in question. The DHCP `Host Name` option is commonly used for identification and to update the address of the host in *local* name services (see Section 2.1). The `Client FQDN` [23] in turn can instrument *global* changes, if the client so desires. An open question is which option devices send identifying info in, why, and whether or not this is used as intended.

A large part of the problem is the practice of dynamically adding `PTR` records. While unique identifiers in `PTR` records enable tracking of specific clients, even record presence in itself provides insights into network dynamics, which combined with other information (e.g., knowledge about building-level IP subnet assignments) stands to reveal a lot. This should be better understood by network operators. Our advice to network operators to reduce the harm of this problem is to block the propagation of `Host Name` information from DHCP to DNS. This can be done by reviewing and adapting the configuration of the internal networks. IPv6 configuration also requires attention. In our study, we mainly focus on IPv4 addresses due to the complexity of efficiently scanning the IPv6 address space at scale. However, an attacker willing to infer information on a specific target can leverage previous studies (e.g., Borgolte et al. [4]) to perform a targeted scan. In addition, when focusing on IPv6, attention should be put on investigating the interaction between domain names and IPv6 addresses when SLAAC and stateless DHCPv6 are in use [20], since this can lead to even more fine-grained tracking of a specific host, thus increasing privacy risks.

9 ETHICAL CONSIDERATIONS

We follow best practices with regards to Internet measurements during our study. Where possible, we rely on existing data sources (Rapid7, OpenINTEL) rather than conducting our own Internet-wide scans. For the supplemental measurement we use to augment the existing data sets, we ensure that we rate limit our scans, we only target address space in networks that we know to be used for dynamic address allocation, we set up a web page on the scanning host that clearly explains the purpose of our study and we act immediately on requests from operators to opt out of our measurements.

IRB Approval. In addition to following the best practices outlined above, our study was reviewed by our institution's ethics board prior to the start of the study. We brought up two concerns in our request to the IRB. First, even though the data we process is publicly accessible and anyone can query it, the purpose of our study and the way we perform targeted analysis raises obvious privacy concerns. Given that our goal precisely is to demonstrate these concerns — that dynamic updates to rDNS data raises privacy issues — this is unsurprising. In order to mitigate this concern, we store our analysis results in compliance with the EU's General Data Protection Regulation and delete data after the research concludes. We note that while this removes the immediate privacy concern of the analysis, in which we pinpoint individual users, the actual privacy threat remains, since anyone can still query the data and reproduce our results. To further mitigate this concern, we do not disclose the names of organisations whose networks we selected for further study, and we take care to report on users in aggregate only. Finally, when zooming in on individual given names, we deliberately pick a very common name.

Second, we sought express permission from our IRB for our supplemental measurement, as this measurement further aggravates the privacy risk by obtaining more timely information about the presence of devices and users on targeted networks. The IRB approved the supplemental measurement, *under the express condition that we minimise the number of networks to which we apply this measurement.* We detail how we select a minimal set of networks to cover with the supplemental measurement in Section 6.1. In the interest of reproducibility, we retain the data from our supplemental measurement in encrypted form on our institution's servers.

Finally, we limit our analysis of specific use-cases (i.e., Life of Brian) to a single name to minimize the possible privacy leak. These analyses can be conducted on any name. However, exploring this goes beyond the scope of our IRB approval.

An approval record from our institution's ethics board is available under registration number *RP 2021-202*.

[8] https://bluecatnetworks.com/

[9] https://www.efficientip.com/

[10] https://www.infoblox.com/

[11] https://menandmice.com/

[12] https://www.solarwinds.com/

10 CONCLUSION AND FUTURE WORK

In this paper, we performed a first of its kind study into the privacy implications of combining DHCP exchanges with dynamic updates to the global DNS. Our findings do not only substantiate existing concerns that unique DHCP client identifiers can carry over to the DNS, but also reveal that reverse records are based on privacy-sensitive information such as client device owner names, and their makes and models. This implies that individuals, and the presence of devices likely belonging to such individuals, are at the risk of being tracked. We analyzed the temporal relation between the presence of rDNS records and the presence of client devices in a network and showed that – for a selection of academic, enterprise and ISP networks alike – records tend to linger for at most one hour after clients leave the network. Via three case studies, we demonstrate that virtually anyone on the Internet can infer and track the presence of specific clients and observe network dynamics via reverse DNS, even with other mechanisms to limit tracking in place. Finally, we began a discussion about the finer details of possible causes and identified ways to start mitigating this problem.

Future work. To aid mitigation efforts, there is dire need for an investigation into how DHCP server and IPAM software carry over privacy-sensitive client identifiers to the global DNS. This is especially problematic if carry-over results from default settings or settings not well understood by network operators. One could investigate from which DHCP option the information is taken (e.g., `Host Name` or `Client FQDN`), if this goes against the intended use of the option, and whether or not client-signalled *desires* for servers not to update DNS records are followed. On the side of DHCP client implementations, one could investigate which DHCP options are filled with *device names*, whether or not this is necessary, and which steps vendors can take to (partly) mitigate the problem. Identifying exposing implementations and client devices could be done in a lab or by using DHCP data collected inside networks.

In this paper we did not aim to exhaustively identify exposed networks with the highest possible accuracy. Rather, we took the first steps towards studying the problem and used *given name* matching as a starting point to drill down in `PTR` records. Future work could go in the direction of investigating the full extent of the problem, by applying techniques from related work to find patterns in hostnames. Another angle could be to consider *forward* DNS data, which can also be dynamically updated by DHCP servers. Future efforts, provided that they are ethically feasible, could also focus on the potential for fine-grained geotemporal user tracking. We used *a posteriori* knowledge on IP subnet allocations in one case study, but related work has established that access network topology can be inferred from hostnames [28]. We imagine that combining such information with client presence inferences can have far-reaching privacy implications. Finally, our findings support that DHCP release messages have an effect on the time rDNS records linger. Future work could study the behavior of clients in this respect: do clients that *can* send releases actually *do* so and is, instead, not doing so a possible defense mechanism?

REFERENCES

[1] S. Alexander and R. Droms. 1997. *DHCP Options and BOOTP Vendor Extensions.* RFC 2132. RFC Editor.

[2] T. Aura, M. Roe, and S. J. Murdoch. 2007. Securing network location awareness with authenticated DHCP. In *International Conference on Security and Privacy in Communications Networks and the Workshops (SecureComm 2007).* 391–402. https://doi.org/10.1109/SECCOM.2007.4550359

[3] C. J. Bernardos, J. C. Zúñiga, and P. O'Hanlon. 2015. Wi-Fi internet connectivity and privacy: Hiding your tracks on the wireless Internet. In *2015 IEEE Conference on Standards for Communications and Networking (CSCN 2015).* 193–198. https://doi.org/10.1109/CSCN.2015.7390443

[4] Kevin Borgolte, Shuang Hao, Tobias Fiebig, and Giovanni Vigna. 2018. Enumerating Active IPv6 Hosts for Large-Scale Security Scans via DNSSEC-Signed Reverse Zones. In *2018 IEEE Symposium on Security and Privacy (SP).* IEEE. https://doi.org/10.1109/sp.2018.00027

[5] S. Bortzmeyer. 2016. *DNS Query Name Minimisation to Improve Privacy.* RFC 7816. RFC Editor. https://tools.ietf.org/html/rfc7816

[6] J. Chabarek and P. Barford. 2013. What's in a Name?: Decoding Router Interface Names. In *5th ACM Workshop on HotPlanet (HotPlanet 2013)* (Hong Kong, China). ACM Press, 3. http://dl.acm.org/citation.cfm?doid=2491159.2491163

[7] W. B. de Vries, Q. Scheitle, M. Müller, W. Toorop, R. Dolmans, and R. van Rijswijk-Deij. 2019. A First Look at QNAME Minimization in the Domain Name System. In *20th International Conference on Passive and Active Measurement (PAM 2019).* 147–160.

[8] Z. Durumeric, R. Wustrow, and J. A. Halderman. 2013. ZMap: Fast Internet-Wide Scanning and Its Security Applications. In *22nd USENIX Conference on Security (SEC 2013).* USENIX Association, 605–620.

[9] Y. Rekhter et al. 1996. *Address Allocation for Private Internets.* RFC 1918. RFC Editor. https://tools.ietf.org/html/rfc1918

[10] S. Groat, M. Dunlop, R. Marchany, and J. Tront. 2011. What DHCPv6 says about you. In *2011 World Congress on Internet Security (WorldCIS 2011).* 146–151. https://doi.org/10.1109/WorldCIS17046.2011.5749901

[11] B. Huffaker, M. Fomenkov, and k. claffy. 2014. DRoP: DNS-based router positioning. *ACM SIGCOMM Computer Communication Review* 44, 3 (2014), 5–13.

[12] C. Huitema, T. Mrugalski, and S. Krishnan. 2016. *Anonymity Profiles for DHCP Clients.* RFC 7844. RFC Editor.

[13] B. Imana, A. Korolova, and J. Heidemann. 2021. Institutional Privacy Risks in Sharing DNS Data. In *Applied Networking Research Workshop (ANRW 2021).* ACM, 69–75. https://doi.org/10.1145/3472305.3472324

[14] A. R. Kang, J. Spaulding, and A. Mohaisen. 2016. Domain Name System Security and Privacy: Old Problems and New Challenges. http://arxiv.org/abs/1606.07080.

[15] Y. Lee and N. Spring. 2017. Identifying and Analyzing Broadband Internet Reverse DNS Names. In *13th International Conference on Emerging Networking EXperiments and Technologies (CoNEXT 2017).* ACM, 35–40. https://doi.org/10.1145/3143361.3143392

[16] M. Luckie, B. Huffaker, A. Marder, Z. Bischof, M. Fletcher, and k. claffy. 2021. Learning to Extract Geographic Information from Internet Router Hostnames. In *ACM SIGCOMM Conference on emerging Networking EXperiments and Technologies (CoNEXT 2021).* 440–453.

[17] M. Luckie, A. Marder, B. Huffaker, and k. claffy. 2021. Learning Regexes to Extract Network Names from Hostnames. In *Asian Internet Engineering Conference (AINTEC 2021).* 9–17.

[18] Paul Mockapetris. 1987. *Domain Names - Concepts and Facilities.* RFC 1034. RFC Editor. https://tools.ietf.org/html/rfc1034

[19] Paul Mockapetris. 1987. *Domain Names - Implementation and Specification.* RFC 1035. RFC Editor. http://tools.ietf.org/html/rfc1035

[20] T. Narten, R. Draves, and S. Krishnan. 2007. *Privacy Extensions for Stateless Address Autoconfiguration in IPv6.* RFC 4941. IETF. http://tools.ietf.org/rfc/rfc4941.txt

[21] OpenINTEL. 2021. OpenINTEL: Active DNS Measurement Project. https://openintel.nl.

[22] Rapid7. 2021. Project Sonar. https://www.rapid7.com/research/project-sonar/.

[23] M. Stapp, B. Volz, and Y. Rekhter. 2006. *The Dynamic Host Configuration Protocol (DHCP) Client Fully Qualified Domain Name (FQDN) Option.* RFC 4702. RFC Editor.

[24] Dennis Tatang, Carl Schneider, and Thorsten Holz. 2019. Large-scale analysis of infrastructure-leaking DNS servers. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment.* Springer, 353–373.

[25] J. Tront, S. Groat, M. Dunlop, and R. Marchany. 2011. Security and privacy produced by DHCP unique identifiers. In *16th North-East Asia Symposium on Nano, Information Technology and Reliability (NASNIT 2021).* 170–179. https://doi.org/10.1109/NASNIT.2011.6111142

[26] R. van Rijswijk-Deij, G. Rijnders, M. Bomhoff, and L. Allodi. 2019. Privacy-Conscious Threat Intelligence Using DNSBLoom. In *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM 2019).* 98–106.

[27] T. Wicinski. 2021. *DNS Privacy Considerations.* RFC 9076. RFC Editor. https://tools.ietf.org/html/rfc9076

[28] Z. Zhang, A. Marder, R. Mok, B. Huffaker, M. Luckie, k. claffy, and A. Schulman. 2021. Inferring Regional Access Network Topologies: Methods and Applications. In *21st ACM Internet Measurement Conference (IMC 2021).* 720–738.