Spectroscopy of Private DNS Update Sources

Andre Broido, Evi Nemeth, kc claffy
CAIDA, SDSC, University of California, San Diego
E-mail: {broido, evi, kc}@caida.org

Abstract—
We study attempts to dynamically update DNS records for private (RFC1918) addresses, by analyzing the frequency spectrum of updates observed at an authoritative nameserver for these addresses. We developed a binary autocorrelation algorithm and discovered that updates come in infinite series with periods of 60 or 75 minutes. We identify both periods as default settings of out-of-the-box Microsoft Windows 2000 and XP DNS software. Identifying this common property of end-user environments helps to understand users' behavior on the Internet. To our knowledge this is the first study of the global impact of dynamic DNS.

I. Introduction

The Internet owes its popularity to a host of intertwined services. Before these services existed, the infrastructure was the playground of a handful of technical experts and university students. Fundamental services of the current public Internet include a search system, a naming system, and a routing and forwarding system. The search system converts a verbal description of a desired information resource to a human-readable Internet domain name. The Domain Name System associates such a domain name with an IP address. The routing and forwarding system allows a packet to travel from its source to a desired destination IP address, subject to the connectivity policy constraints of Internet service providers (ISPs).

All of these systems were designed for traffic loads that reflect the rate and complexity of human activities. At these rates a few high end workstations can conceivably provide the whole population of Internet users with a specific service. However, services that may be robust in the face of human-triggered request behavior can be easily overwhelmed when streams of machine-generated requests converge on a few servers, even when each individual machine produces just a trickle of traffic.

In this paper we analyze one kind of such spurious traffic in the worldwide Domain Name System (DNS) – attempts to erroneously, and incessantly, update address-to-hostname mappings for private addresses in nameservers at the top of the DNS hierarchy. We found large fractions of this traffic to be repetitive and periodic.

Most of the DNS traffic we observed dealt with so called private, or RFC1918 addresses. We discovered that a large portion of these updates is caused by the default configuration of the DHCP/DNS servers shipped with Microsoft systems. This server software sends periodic updates with frequencies that we found with spectral analysis and confirmed via two methods: a laboratory experiment and vendor documentation. This (mis)configuration is so widespread that patterns of Internet access by end users are reflected in the pulsations of the flow of DNS updates. We call these pulsations the *heartbeat of private networks* – privately owned and using private addresses. There is no reason to believe that these spurious DNS packets are limited to private networks; our data sources primarily reflected attempts to update data about private address space at the root nameservers. (Due to space limitations we assume familiarity with the basics of the DNS; see [1], [2], [3].)

RFC1918 updates are part of a general problem with some (in this case, Microsoft's) software or its configuration, whereby the content of local variables (in this case, DNS names for RFC1918 addresses) escapes the local environment and 'leaks' into the public Internet and in particular to the root name servers [4]. Users are not aware of the fact that their machines are misbehaving. The resulting traffic is not only a waste of global Internet resources, public and private, but also raises security, privacy and intellectual property questions [5].

Since mid-2002 almost all RFC1918 spurious update traffic is deflected from the root nameservers to a constellation of *blackhole servers*. These dedicated servers do not solve the underlying problem but at least partly protect the root servers from the misguided traffic. (Update-related SOA queries still reach root servers.) They also facilitate a focused analysis of the dynamics of DNS updates. In October 2002 we observed an alarming trend in the continued growth (doubling in four months) of RFC1918 update traffic at one of the blackhole servers, surging to over 1300 updates/sec at midnight November 02. This increase might have been caused by the back-to-school timeframe, OS upgrades to Windows 2000 and XP, or routing idiosyncrasies. Growth stabilized in November when two blackhole servers colocated with `k-peer` and `i-root` became fully operational. We do not yet completely understand the phenomenon of private DNS updates, in particular how it may affect the root servers if the blackhole servers succumb to the pressure of RFC1918 traffic.

The performance analysis we present here extends work on measurement, performance and placement of DNS root servers [6] [4] [7], and on the use of private and unrouted addresses [8]. In particular, [4] and [9] [10] discuss the pervasiveness of DNS misconfiguration as observed in queries

Support for this work is provided by the Defense Advanced Research Project Agency (DARPA), through its NGI (N66001-98-2-8922) and NMS (N66001-01-1-8909) programs, and by the National Science Foundation (NSF NCR-9711092). CAIDA is a collaborative organization supporting cooperative efforts among the commercial, government and research communities aimed at promoting a scalable, robust Internet infrastructure. CAIDA is based at the University of California's San Diego Supercomputer Center (SDSC). www.caida.org.

reaching the root servers. An abstract of our study can be found in [11]. Liston *et al.* [12] provide a complementary view of the root server load problem.

In addition its practical significance, our interest in RFC1918 DNS updates is motivated by the research value and conceptual richness of the data originated in private nets. It presents wide coverage of the Internet's periphery of which previously only scattered glimpses were available. It intertwines fundamental services of DNS, DHCP and NAT. It arises from the deflection of root sever traffic achieved by clever application of the semantic constructs of DNS (authoritative service for reusable addresses) and BGP (anycast routing of the update traffic.)

Techniques that search for delay quantization patterns apply to a variety of other kinds of traffic. We have used them for bitrate estimation and broadband source identification [13]. We expect in the future to apply them to analysis of the periodicity of BGP updates, which complements the results of [14] and [15].[1] In general, we observe that the approach of *network spectroscopy*, i.e. object identification by spectra of periods and delays [17] [18] [19] is rapidly becoming a prominent method in Internet science.[2]

The rest of the paper is organized as follows. In section II we discuss the reasons behind the use of RFC1918 addresses for private networks and give background on the AS112 project that offloads RFC1918 update traffic from the root servers. Section III describes our data sources and preliminary observations on the magnitude of the spurious update problem. All these observations suggest that the updates are generated by home and small office computers from all over the world. Sections IV and V examine the update flows from individual sources and analyze the spectrum of update arrivals. Section VI describes the results of laboratory experiments with Windows 2000 and Windows XP nameservers, which confirm the periodic nature of updates generated by off-the-shelf Windows software; we quote Microsoft documentation explaining that this is exactly the vendor's intent. Section VII contains conclusions, directions for future work and suggestions for default configuration changes to ameliorate some of the damage.

II. Private addresses

The explosive demand for Internet addresses threatened the depletion of the IPv4 address space in the mid-1990s. In response, the Internet Assigned Numbers Authority (IANA) decided in 1996 to allocate a portion of the address space for use in private networks. These so called RFC1918 [21] addresses can be used without coordination with IANA or an Internet registry. The RFC1918 policy allowed for local reuse of certain addresses which meant that the IANA could be more conservative in allocation policies for globally visible addresses. CIDR [22] (Classless Inter-Domain Routing) also made it possible to use the address space much more efficiently. These three factors dramatically slowed down address consumption [23] for the next decade and potentially increased the lifetime of IPv4 for several decades.

RFC1918 specified three address blocks for private use[3], 10/8 with 16.8 million addreses, 172.16/12 with 1 million addresses, and 192.168/16 with 65536 addresses.

RFC1918 and other private addresses are not intended to be unique and therefore are not globally reachable (or routable). Networks that use these addresses may get their global connectivity through a bridging device that, transparently, maps internal addresses to globally unique ones for the purposes of communication with the outside world. Such network address translation (NAT) [24] is used extensively at the edges of the Internet, in households or small businesses that connect more than one IP device via telephone, DSL or cable modems.

The authors of RFC1918 warned of locally scoped addresses leaking into the global Internet:

"Indirect references to such addresses should be contained within the enterprise. Prominent examples of such references are DNS Resource Records and other information referring to internal private addresses. In particular, Internet service providers should take measures to prevent such leakage."

In this paper we examine how the above tenet is violated today and observe that millions of DNS packets are sent daily to nameservers outside private nets requesting or containing information on RFC1918 addresses. The fact that RFC1918 addresses may have locally valid DNS information means that DNS software cannot categorically ignore transactions involving such addresses without throttling the operation of otherwise properly configured internal networks.

An RFC1918 address may appear in DNS packets as either the source address of the packet or as part of the DNS payload inside the packet, i.e., the subject of a query or update. In the first case there is no route back to the sending host and the packet cannot be answered at all, not even to notify the source of the erroneous traffic. In the second case the sending host has a valid IP address but these RFC1918 address mappings are irrelevant for the servers outside the local net of the sending host.

A. Dynamic DNS updates

To allow for reuse and conservation of IP addresses within a given site, IP addresses are often obtained dynamically using the DHCP (Dynamic Host Configuration Protocol) [25], [26]. For example a dialup modem may lease an IP address from the ISP's DHCP server as a part of the initial connection establishment. This address is guaranteed to be unique in the local context. DHCP servers periodically renew host address leases. If the host has a DNS name, this address (re)assignment via DHCP may alter the DNS mapping between the hostname and address, making

[1] A clustering algorithm based on correlations of update time series was recently presented in [16].

[2] "Although the Internet is an engineered artifact, it now presents us with questions that are better approached from a scientific posture" — Mark Crovella [20].

[3] Two other reserved blocks, 169.254/16 (link-local) and 192.0.2/24 (test-net) are considered private; test-net does not appear in the logs we analyzed.

the existing DNS records inaccurate. Many DHCP/DNS implementations, including reference implementations by ISC (Internet Software Consortium) and Microsoft's Windows 2000 and XP, have the ability to send dynamic DNS updates [27].[4]

B. RFC1918 DNS servers: AS112 project

In [4] we found that since the release of Windows 2000, DNS update packets for private address space leak from local intranets and reach the root servers – the top of the Internet naming tree. The root servers refuse these updates and log an error. As the imposed load on root servers increased, separate authoritative servers were deployed in several locations [28], [29], [30] to protect the root servers from this illegitimate traffic. These servers have substantially reduced the spurious update load on the root servers.

The implementation of this solution relied on the use of *anycast* [30] routing. An anycast address refers to a group of machines that respond identically as a single server but are at multiple topological locations across the Internet. Each server handling RFC1918 PTR query or update traffic uses a well-known address from an allocated IPv4 address block (192.175.48.0/24) which is announced in global BGP routing tables as belonging to autonomous system (AS) 112. The address 192.175.48.1 (`prisoner.iana.org`) handles RFC1918 updates; .6 and .42 (`blackhole-1.iana.org, blackhole-2.iana.org`) handle RFC1918 queries.

As of April 2003, authoritative servers respond to these anycast addresses from the US, Japan, Brazil, Britain, Denmark, Sweden, The Netherlands, and Bulgaria.[5] The routing system routes to the 'nearest' instance of the anycast address.

III. Measurements

Our data is obtained from `hazel.isc.org`, an instance of an authoritative server for RFC1918 addresses that is located near F-root in Palo Alto, California. We use log files collected between May and October 2002. For a packet to appear in `hazel`'s log files it must be a DNS packet carrying an update for an RFC1918 address to hostname mapping that has leaked from a client's local subnet.

We analyzed three datasets: *D0* on 16 May 2002, *D1* from 28 May to 4 June 2002 (7 days) and *D2* from 4-30 July 2002 (26 days).

A. Updates per RFC1918 block

We examined the attempted updates per address block[6] during a 3.5 day period in our May-June data set. We saw 35 M (68% of total RFC1918) updates to addresses in 192.168/16, 12.4 M (24%) to 10/8, and 3.8 M (7.5%) updates in 172.16/12. IP addresses in the old ARPAnet range (10.0.0.0/8) are often used in corporate environments, e.g., VPNs. This space is often managed by professional system administrators resulting in a lower rate of address leakage. The 192.168.0.0/16 block is often used by manufacturers of networking gear for home and small office use: NATs, firewalls, DSL routers, etc. These devices have either manufacturer defaults that assign 192.168.0.0/16 addresses to local hosts behind them, or advise users via product documentation to configure such behavior. Windows software (2000, XP) includes a mini-DHCP server that does ICS (Internet Connection Sharing), Microsoft's name for NAT [24], and uses private addresses in 192.168/16 by default [5]. The block 172.16.0.0/12 is not as popular and generates fewer RFC1918 updates. It is used by some universities [31] for internal routing, mostly for security reasons.[7] Since temporal variation in update traffic appears independent of the particular RFC1918 address block, we will characterize (in Section IV) the temporal aspects of the aggregate RFC1918 workload rather than by individual blocks.

B. Updates per source IP

In this section we classify RFC1918 update attempts by layer 3 attributes such as IP address, port, network prefix and autonomous system (AS). A small fraction (1%) of update attempts in our data are *from* RFC1918 source addresses and so cannot be attributed to a single host. They are neglible and do not affect our statistics of host counts and per host averages. Such RFC1918-sourced traffic is filtered out by the routers' access control lists before reaching the DNS server.

Figure 1a shows the distribution of update counts by the first byte and first two bytes of the source IP address. The bands of points correspond loosely to IP address allocation boundaries. The plot shows two granularities, indicated by black squares and grey dots. The black squares show the total number of updates for each /8 segment of the IPv4 address space. The grey dots show the number of updates for each /16 block (smearing) within the enclosing /8.

The largest individual /8 contribution comes from the 24/8 block, mostly used by cable modem providers. Many recent allocations with first byte between 60 and 68 also belong to broadband end user connectivity providers, whose customers use RFC1918 addresses behind NAT modem/routers. The prominence of RFC1918 updates coming from those blocks may reflect the fact that providers often charge customers for additional IP addresses, which motivates the use of NAT boxes at homes and small businesses.[8]

Table I shows the number of IP addresses in our sample that come from each of the traditional Class A, B, and C

[4] In the Microsoft nameservers these dynamic updates are turned on by default.

[5] RouteViews (56 tables) of April 01, 2003 show 10 ASes: ISC, Verisign, WIDE, LONAP, New Mexico NAP, AT&T, RIPE, TeleDanmark, Netnod, Port80, Any site wanting to run servers to confine RFC1918 updates to their own networks should contact the AS112 group [28].

[6] The log files contain entries for each /16 network in the 172.16.0.0/12 block; we aggregate results over the whole /12 block.

[7] We also see a few (approximately 1 in 500 updates and 1 in 300 source IPs in the D1 dataset) update requests for the link-local address block, 169.254/16. Link-local addresses are assigned when DHCP fails and no other address is configured; they are never routed or forwarded beyond the local network and should not be configured in the DNS [32] [5].

[8] In October 2002 Pacific Bell's [33] basic DSL service with one dynamic IP costs $50/month and with 5 static IPs costs $65/month.

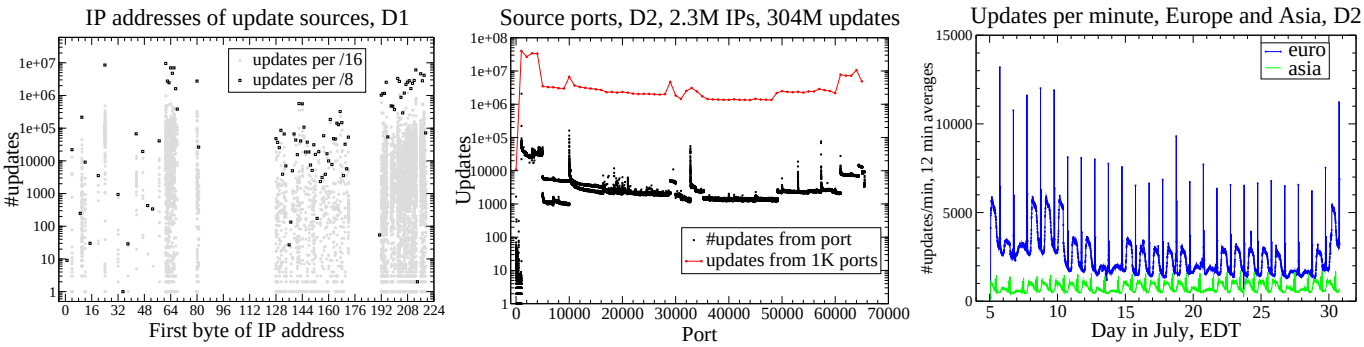

Fig. 1. a) Source addresses of update traffic, D1 b) Source ports in updates, D2 b) Diurnal variation for Europe and Asia, D2

TABLE I
UPDATES PER IP ADDRESS CLASS, D1

Class	IPs	Percent	Updates	Percent
A	507541	42.2	47263887	48.2
B	65177	5.4	3048519	3.1
C	631432	52.4	47787751	48.7
Total	1204150	100.0	98100157	100.0

ranges. These statistics agree with Figure 1 and confirm that Class B networks (first byte between 128 and 191) are rarely the source of RFC1918 updates (only 5% of unique IP addresses and 3% of updates in our data set are from class B sources). Traditionally class B space was allocated to universities and medium sized businesses. Many class B allocations happened before allocations in class C space and the upper half of class A space.

C. Updates per source port

We examined the source ports on update packets to help identify the operating systems responsible for the spurious update traffic. In some cases the sequence of ports used by a machine (and the point in the port space where the sequence wraps) identifies the operating system on the sending machine. Solaris uses ports 32768-65535, FreeBSD 5.0 49152-65535, and older BSD, FreeBSD 4.0, Linux 2.4 and Windows 2000 use port range 1024-5000 (some of them exclude one or another end of the range.) As we will see below, update attempts use both UDP and TCP.

Figure 1b shows that the port space usage is not uniform but rather concentrated in the range 1024-5000 (44.3% of all updates) with a lower peak close to the end of the possible port range. Windows and older BSD-based systems use that range for ephemeral ports. Port 1025 was the most frequently used source port across all updates seen in the datasets.[9]

D. Continents and diurnal patterns

We used ARIN tables of allocated address blocks for 1 April 2002[10] to map IP source addresses to continents. This data includes allocations from other regional registries (RIPE, APNIC) as well. We mapped all IP addresses that were the source of RFC1918 update attempts to their respective countries of origin and continents. This placement method may be inaccurate in a few cases where companies are registered in one country but have IP-addressed equipment in another.[11] Table II shows the breakdown of updates by registry area. Unknown addresses include RFC1918 sources and IP addresses not found in registry allocations. The message is clear: illegitimate updates are a global phenomenon that is not confined to or dominated by a specific market or continent.

TABLE II
HOSTS AND UPDATE ATTEMPTS BY CONTINENT, D1

Region	Hosts	Percent	Updates	Percent
America	327616	27.2	49029151	50.0
Asia	372974	31.0	25041172	25.5
Europe	484227	40.2	22314423	22.7
Unknown	19345	1.6	1541059	1.6
Total	1204162		97925805	

Figures 2a 1c, shows North and South American[12], European and Asian patterns of diurnal and weekly variation in the stream of RFC1918 updates. The data is a mixture of singular spikes and smooth periodic patterns. Large spikes of updates occur near midnight in time zones of significant Internet user population. We see four in America an hour apart with the largest at east and west coasts (Fig. 2b). Graphs for Asia show three spikes; graphs in Europe (Fig.1c) show two spikes, one in Britain and another in the continental Western Europe. A detailed view of one of the large spikes (Fig. 2c) reveals a plateau immediately after midnight where a quadruple load (compared to the baseline) lasts for about 2 min.; double load lasts for almost 6 min. Many systems are sending an update at midnight but unsynchronized clocks serve to dissipate this surge over about six minutes. Since more precise synchronization could overwhelm the system, these unsynchronized clocks mitigate a potential catastrophe at the root servers. Ignoring the large spikes, the smooth patterns closely resemble daily patterns of individual activity, where updates occur when people turn on their computers.

[9]Parallel strips in 10,000 range are caused by higer frequencies of even-numbered ports.

[10]ftp://ftp.arin.net/pub/stats/

[11]There are alternative methods, e.g., CAIDA's *netgeo* [34] or related commercial mapping tools, but we did not find significant differences in results among the methods.

[12]All ARIN address allocations, including South Africa. US and Canada accounted for 96% of ARIN address blocks.

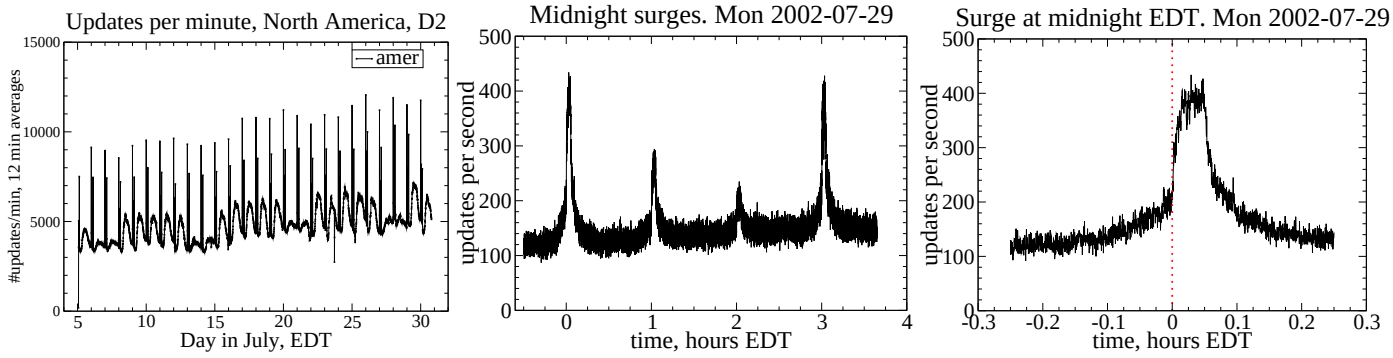

Fig. 2. a) Weekly fluctuations in update rate, US b) Surges at US midnights c) US East Coast midnight surge

E. AS contributions

Each network that uses BGP for routing on the Internet has an AS (Autonomous System) number to identify it in interdomain routing negotations. We converted source IP addresses to AS numbers using the University of Oregon's RouteViews BGP tables [35]. In the May 2002 dataset (D1) 3309 different ASes tried to update the RFC1918 records; the top 20 of them (0.6%) account for half of the updates. These 'elephant' contributors dominate AS-level update flows.

The largest numbers of updates come from incumbent telecom carriers for various regions: Chinalink (China, 7.5%), Pacific Bell (US, 6.4%), Ibernet (Spain, 6.3%), China Telecom (5.5%), Southwestern Bell (US, 4.67%), Telus (Canada 2.2%), and Hong Kong Telecom (1.5%). Cable companies Cablevision, Adelphia, Time Warner/RoadRunner are also among the top twenty. Countries such as China who were relatively late in deploying extensive Internet infrastructure have more difficulty getting enough global address space allocated from registries and therefore tend to use RFC1918 address space.

At the granularity of individual IP address counts rather than update counts, the top 20 ASes contain over 54% of all IP addresses from which updates were sent. The list again has Chinalink and Ibernet at the top but also includes a few new players,[13] mostly ISPs serving home users and small businesses. Note that these ASes reflect the location of the F-root server, closer to Asia than many other root servers. We see that both IPs and updates come from ASes that serve large populations of users; academic and corporate networks are underrepresented in that list.

F. Source domain names

To analyze domain names associated with IP sources we used the May 16 (D0) dataset, and extracted components of the domain name, discarding top level country codes and com, net, and org suffixes. Half of all source IPs belong to just 23 dialup, DSL and cable modem provider domain names. Looking from a different angle, we find that words associated with end users:

{ `catv, cable, client, cust, dial, direc, dsl, host, hsia ("high-speed Internet access"), nat, online, pool, port` }

are present in 113847 (51.2%) of the DNS names. Many DNS names contain an encoding of an IP address, often autogenerated rather than registered on an individual basis. Domain names are another bit of evidence that updates come from end-user machines.

G. Updates at the root servers

To assess the current magnitude of the problem at root nameservers, we also analyzed a `tcpdump` trace taken at the `k-root` (London) on Aug.22, 2002. The trace contains 8.0 M DNS packets, of which 4.1 M are queries. Less than 1000 packets (0.02% of the total) represent updates. Some of the repeated updates are separated by intervals of 5 or 10 min, Most of these updates did not contain RFC1918 addresses. Similar statistics were observed at several other root servers, including `m-root, i-root` and `e-root`, in traces of the same duration taken on the same date.

The fact that the fraction of updates observed at the root servers is so tiny suggests that:

1. DNS updates that leak outside local networks consist mostly of RFC1918 updates.
2. AS112 blackhole servers capture almost all RFC1918 DNS updates.

The impact of update-related traffic on root servers, however, is much higher than the root servers' update counts would suggest. Before a dynamic update can occur, the primary nameserver for the domain being updated must be determined by an SOA query. Our analysis showed that the majority (69%) of the SOA queries at `k-root` were in preparation for a dynamic update. This is 6.3% of the total traffic at `k-root`.

IV. Flow durations and sizes

A. Mice, mules and elephants

Our observations in the previous sections suggest that most DNS RFC1918 updates come from small user environments. We now turn to analyzing the nature of individual update streams and identifying which platforms and operating systems contribute the bulk of the updates.

To determine where update attempts originate, we calculate the relative importance of sources with various contribution sizes. We have seen that on the AS level the

[13] Swisscom IP-plus (Switzerland), NTT Communications (Japan), Energis Squared (UK), TELEKOM-AT (Austria), Arcor (Germany), France Telecom, EarthLink (US) and Planet Media (Netherlands).

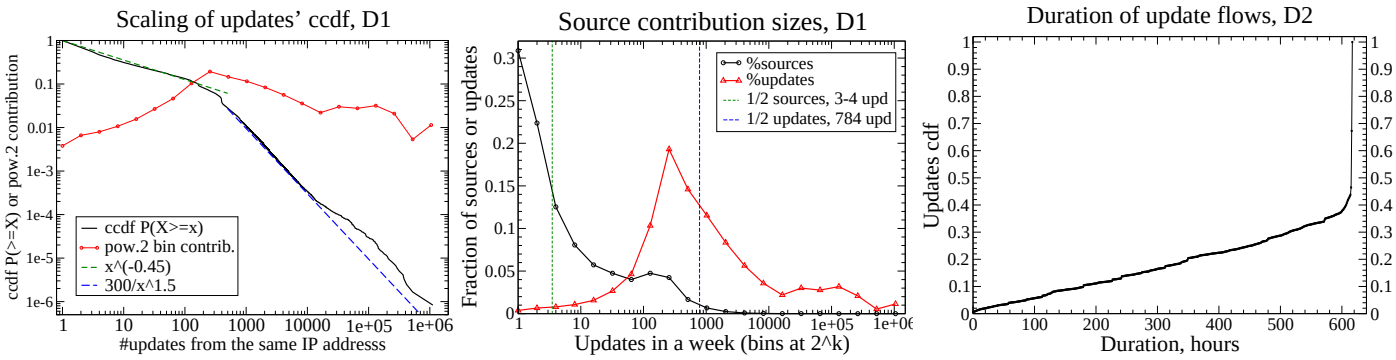

Fig. 3. a) Scaling of updates' ccdf b) Mice, mules and elephants c) Update flow duration cdf

majority of updates comes from a handful of participants. It would be useful to know if a few hosts are responsible for most of the observed traffic. We next calculate the distribution of contributed update counts per source. We distinguish among hosts that individually contribute high workloads (elephants), hosts that generate small workloads but are present in extremely large numbers (mice), and hosts in between (mules).

Size distributions in Internet statistics are usually described by ccdf, which measures the number of objects exceeding the value on x-axis. To handle extremes of large sizes and small chances, ccdf's are usually presented in log-log plots as in Fig. 3a, which shows scaling of approximately $x^{-0.5}$ in its initial portion, then drops and crosses over to $x^{-1.5}$. This description, however, does not tell which part of the size spectrum contributes most updates.

We will estimate the relative importance of each group by separating host contributions to total traffic workload into bins (by powers of 2) and evaluating the number of hosts and amount of traffic in each bin. Figure 3b shows the update contributions; dashed vertical lines mark the middle of the distribution, with half of the sources (or updates) on each side of the line.

For the weekly log from 28 May-4 Jun 2002 (D1):

- Half of all sources send 3 or fewer updates (mice)
- Half of all updates are sent by sources with ≥ 784 updates (1.6% of all sources)
- The largest single source contribution is over 1 million, or 1200 $\times$ the median (elephants)
- The mode, or largest single-bin contribution (20%), comes from sources with 256-512 updates (mules)

1.6% of the sources contribute half of the updates at the IP granularity, whereas only 0.6% of the ASes contribute half of the updates. The average (arithmetic mean) is 81.5 updates per source IP address. 95% of the updates come from 20% of the sources. This is also relatively large compared to other cases of Internet traffic volume disparity [36]. Thus in our sample the workload is dominated by midrange contributors – mules, rather than elephants or mice.

B. Long-lasting flows contribution

We will now analyze the persistence of individual hosts in attempting to update reverse RFC1918 DNS records. We call the stream of update packets from a particular IP address to our authoritative servers a *flow*.

Figure 3c shows the duration of update flows, that is the intervals of time over which individual source IP addresses were observed in the 26 day period (624 hours) in July 2002 (dataset D2). The x-axis is the time between the first and last update from a source IP address, even if this source was inactive in the interim. The y-axis is the cumulative update fraction (CDF), i.e., the fraction of updates in flows with duration $t \leq x$. About 60% of the updates came from hosts that were updating for the entire measurement period (the knee of the function). The remaining 40% of updates have about a uniform chance of being in a flow of any smaller duration. The distribution of updates is close to the sum of a uniform distribution and an atom of probability at the full duration. Its shape is similar to the distribution of persistence of Internet IP level paths in [37].

There is an uncertainty in identifying a host by IP address only, especially in dialup ISP networks, where hosts frequently reuse IP adresses from the same pool. Possible remedies such as identifying hosts by port range and specific RFC1918 block, using timeouts for update flows, or the IP ID field [38], require further investigation.

V. Spectral analysis

We now turn to the nature of the update flows and explore their similarity and time-dependent behavior. Our datasets include a timestamp with millisecond granularity.

A. Interarrival times

Knowledge of interarrival times is important for modeling the flow of DNS updates. A well developed mathematical theory deals with Poisson processes whose interarrival time distribution is exponential (see [39] and references therein). The equations for Poisson processes admit simple analytic solutions for expected queue size and service time. In general these processes represent the simplest model for a flow of events that occur independently, at random, and with a constant average arrival rate.

We found that the body of the distribution of interarrival times for all updates (viewed as one stream) was close to exponential especially over short accumulation intervals, up to several hours. However this exponential model is only valid for the highly multiplexed aggregate stream of updates coming from diverse sources. As we will show in the next section, individual sources are often periodic.

The distribution of interarrival times for the dataset D2 (26 days) shown in Figure 4 is close to exponential for interarrival times up to 0.1 sec, where the probabilities are as small as 10^{-5}. It becomes a long-tailed distribution close to a power function for durations longer than 0.4 seconds. (The largest interval we saw in 26 days was 64 seconds.)

B. Update periods

Despite the exponential nature of the aggregate stream of updates, many individual update sources exhibit regular periodic patterns. We have already discussed the spikes positioned at midnight for various time zones. Other patterns arise from individual update sources with periods of 75 minutes, 1 hour and short periods under 1 minute.

To see how many update sources are periodic, we examined average update rates for sources present in the 26-day July dataset.[14] Figure 4b shows the density of updates versus the update rate with a resolution of 20 bins per decade. We took only sources whose update series lasted longer than an hour, resulting in the removal of 882,633 sources (1,582,417 updates) leaving 1.45M source hosts with 302M updates over 26 days in July 2002. The solid line is updates and the dashed line source IP addresses.

The two large spikes in Figure 4b represent periods of 60 and 75 minutes. 5% of the updates come from sources with average update rates in the range 1-1.122 per hour (60-minute cycle) and 8% from sources with 2.24-2.51 updates per hour. This 8% actually matches a cycle of 3 updates in 75 minutes. The next noticeable spike is at twice this rate, most likely caused by networks with two hosts in RFC1918 space, for which 6 updates are generated in 75 minutes and attributed to the same IP address due to NAT mappings.

As the dashed line (percentage of IP sources) shows, most IP addresses are sending updates at much lower rates; half of the sources are sending at a rate of 0.09 or fewer updates per hour. However, half of the updates come from sources sending at rates of 5 or more per hour. The rates of 1 per hour and 3 per 75 minutes account for 6.43% and 3.53% of all observed sources, respectively. Neither of these numbers, however, reveals how strict or loose the periodicity is, nor the spacing of updates within a period.

It is difficult to determine the precise period of updates because sometimes an update is missing from the series, either because a host is switched off, a DNS packet is lost in the network or some activity on the source network interferes with updates. Often an extra sequence of updates is interleaved in the series because another host becomes active on the private network and a local NAT merges its traffic with the original source. For that reason we could not use a Fourier transform on update arrivals to extract a period; lack of coherence in update arrival times defeats the amplifying properties of the transform.

We tested two approaches to finding the update period; both of them use a binary autocorrelation function. By determining the lag (shifts) at which the autocorrelation is maximal, we can find how many updates constitute a period. We then recover the actual (temporal) period from the original interarrival times.

Algorithm 1. We sorted each logfile[15] by the IP address of the source of the update packet, and only used sources with 15 or more updates. We then computed sequences of update interarrival times for each source, rounding them to whole minutes. We then took this sequence, shifted it along itself by 1,2,...,9 positions, and if more than 90% of the interarrival times matched for some shift we classified that source as periodic. A totally periodic source with a simple fixed period would have all interarrival times equal to the same value and the sequence of values would match perfectly if shifted by 1 along itself. The amount of shift necessary for the sequences to match determines the number of updates in a period.

We applied Algorithm 1 to a 7.5 hour logfile from early Wednesday 29 May 2002 that contained 4.67 M updates and 240 K source IPs. Of those, 78 K sources sent 15 or more updates over the duration of the log, of which 32K (40%) sources were found to be periodic. Among the periodic updates, 2001 (6%) have a period of 60 minutes, 22333 (70%) a period of 75 minutes and 2.7% a period of 30 minutes.

In the whole set of 21 logfiles from D1, 38-56% of the sources were periodic. Of these, 1.4-3.5% had periods under 1 minute, 2.6-3.5% 30 minutes, 5.8-12% had 60 minute periods and 64-70% 75 minute periods, and 1% had a 76 minute period.

This approach discovers a smaller percentage of periodic sources when run over the whole one-week dataset D1 because DHCP and/or NAT do not always assign the same IP address to a host after it is turned off, e.g., overnight. In this case the IP address does not uniquely identify a source. Among 315K sources with 15 or more updates, 86580 (27.5%) are identified as periodic. 32456 (38%) of these sources have a period of 60 minutes, 37575 (43%) 75 minutes, and 5503 (6.4%) 76 minutes. The significant drop in the fraction of 75 minute periods is most likely caused by occasional missing updates and/or rounding errors when converting intervals to whole minutes. Both factors distort the periodicity of minute counts.

Algorithm 2. To avoid these inaccuracies we tried a more robust algorithm, which finds the fraction of periodic updates from one source as follows (see also [11]):

1. For all sources with ten or more updates, take the sequence of interarrival times expressed as integers in milliseconds.
2. Convert them to logarithms base two truncated to integer parts.[16]
3. For each shift of the update sequence by 1, 2, ..., 30 updates, count the number of positions in which truncated logarithms in the original and shifted sequences are equal.
4. Find the lag (shift) at which this overlap is maximal; discard the source if the maximal count is less than 10% of

[14] An average update rate is the number of updates from given source minus one, divided by the timestamp difference between last and first update in the series.

[15] In May-July data (D0-D2) an update logfile usually covers about 8 hours and contains up to 5M updates.

[16] We add 1 to integers to disambiguate them from 0 ms.

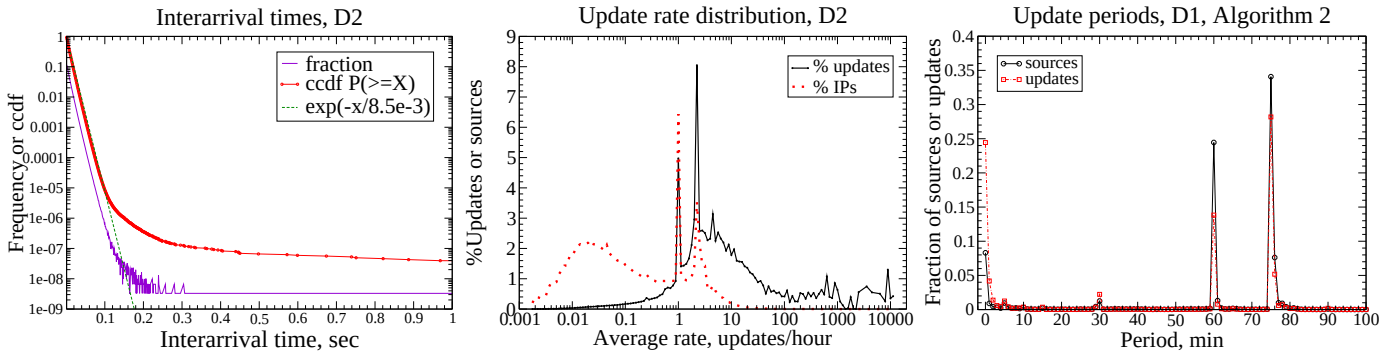

Fig. 4. a) Update interarrival times b) Update rates c) Update periods, Algorithm 2

all its updates.

5. Find the longest contiguous stretch in which every entry equals its shifted counterpart. Discard an IP if this stretch is shorter than the shift (lag.)

6. Extract the interarrival times from the beginning of this longest stretch. Take the sum of these times as the period.

Although more complex, this algorithm worked well and was able to disambiguate interleaved sequences. The update data contains interleaved sequences sent on behalf of several local hosts that can join and leave the private network at arbitrary times. Together with the occasional missing or extra updates, these varied components require a robust algorithm to distill. Clock skew in the source hosts also contributes to noise that must be filtered out, which is why we matched binary logarithms of data rather than numeric values, and relaxed the threshold condition for a source's periodicity (matching 10% of the updates as opposed to 90% in the first algorithm.)

Fig. 4c shows the number of IP source addresses that send a significant fraction of the updates in periodic intervals and the number of updates produced by these sources. 360710 source IP addresses (out of 1.2 M) were included in the analysis; each source contributed at least 10 updates. Together, these sources contribute 97% of all updates. Out of those, 311 K addresses had a contiguous periodic portion at least as long as the identified lag; 87 M (89%) of updates come from these IPs. The largest observed period was 75 hours.

The pattern of the 75-minute update cycle is especially revealing; it usually involves three updates, at intervals of 5, 10, and 60 minutes. It appears as if an attempted update (at "0" minutes) is repeated after a timeout of 5 minutes and then again after doubling the timeout to 10 minutes, after which the system falls back to a default of 60 minutes. Microsoft's documentation (quoted below) lists this as the intended behavior. There is also a strong spike reflecting a sequence of 60 minute intervals. The most frequent periods are: 60 min (24% sources, 14% updates); 75 min (34% sources, 28% updates); and under 1 min (8% sources, 24% updates.) These and nearby periods account for almost 75% of the sources and updates.

Table III lists the most frequent cases of periodicity. The first entry is the 3-part 75 minute period; the second is the single 60 minute period. Combining these sources with those having periods under 1 minute per update, 76 minutes per 3 updates, 75 minutes per 6 updates and 60 minutes with 2 updates, accounts for 2/3 of all periodic IPs. Next in importance are (minutes, updates) pairs of (120,2), (0,2), (75,9), and (60,3) minutes and updates (1-1.5% IPs each). The most frequent update counts in the periods are 3 (43% IPs), 1 (30.5%), 2 (10%) and 6 (7%), followed by 9, 4, 12, 18 and 5.

TABLE III
NUMBER OF UPDATES PER PERIOD, D1

Period, minutes	75	60	0	76	75	60
Period, updates	3	1	1	3	6	2
IPs , K	86.5	58.7	19.2	19.1	11.0	10.4
%Periodic IPs	27.8	18.9	6.2	6.1	3.5	3.3

Periods that include more than one update can arise from two mechanisms. First, several computers can be multiplexed onto one IP source address by NAT. Second, the period identified can be an integer multiple of a true period, e.g. $120 = 2 \cdot 60$ min. (This problem is present in many spectral analysis algorithms [13].)

To estimate the impact of period doubling, we computed how often the first k updates in a period of $2k$ have their duration d_1 within 0.49-0.51 of the whole period's duration d. We found that among 311 K periodic sources, 66.6 K (21.4%) have even-numbered periods (2, 4, 6... updates). These sources originate 30.8 M (31.4% of all) updates. The sources with $0.49 \leq d_1/d \leq 0.51$ (suspected doubled periods) account for 18 K (27%) IPs and 11 M (36%) updates.

The period divisible by 3 updates was present in 173 K sources which sent 48 M updates (55% of 87 M updates from periodic sources.) Out of those, 8603 (5%) source IPs had the first 1/3 of their inter-update intervals adding up to within $1/3 \pm 0.01$ of the periods' duration. These sources contributed a total of 5.85 M (12%) updates. 38 K IPs had periods of 6, 9,.. updates; among them, potential triple periods accounted for 10% of those source IPs.

These observations allow us to conjecture that while a signficiant number of periods may be doubled, tripled etc. by our algorithm, most sources with $2k$ or $3k+3$ updates in their period belong to the networks with several computers. We leave precise identification of these setups for future work.

VI. OS fingerprinting

We have already made an attempt at OS fingerprinting when we collected statistics of ports from update sources in Sect.III. However, this approach could not differentiate between BSD and Windows-based machines.

We next used Ofir Arkin's fingerprinting utility Xprobe 1 [40] on a list of 413 IP addresses collected on 12 July 2002 to identify the operating systems sending DNS updates to the root servers. Since the majority of the traffic seems to be from home and small office connections where computers are turned off when not in use, IP addresses were probed in real time as soon as they appeared in the logfiles. Despite that, the OS breakdown returned by Xprobe was inconclusive. Microsoft Windows boxes were the most prevalent, but several flavors of UNIX/Linux were also present. Devices such as bridges, switches and home routers were also detected, not surprising since most IP addresses in our sample are those used by NATs and firewalls for which special hardware and/or UNIX boxes are frequently used. However, this data does not indicate which OSes are used behind NATs, since Xprobe cannot test private addreses.

Notably missing from the list were Apple systems that, through MacOS 10.1, do not support DNS dynamic updates. However, because our OS fingerprinting efforts did not yield a sufficiently refined picture of the sources of DNS update attempts, we next designed a test network to measure the sources and regularity of update attempts.

A. Test laboratory

Due to the numbers of hosts involved and the fact that they are predominantly from home or small business computers via xDSL or cable modem connections, we suspected that more of the update traffic came from Microsoft Windows boxes than was indicated by Xprobe. To investigate further, we designed a laboratory experiment where we installed PCs running 'out-of-the-box' Win2k desktop, Win2k servers with and without Active Directory, and WinXP operating system software. We also built machines at various patch levels. We captured packet traces for all traffic on this test network over several weeks between November 2002 and January 2003. Most of the analysis here is from a 99 hour trace of taken December 6-10, 2002.

The traces showed the Windows machines sending DNS update packets to the nameserver configured by DHCP at regular intervals. The update sequence was periodic with bursts of packets on 5, 10, 60 and 75 minute boundaries as seen in the data arriving at the RFC1918 reverse zone's authoritative nameserver. Several back-to-back packets were observed at each period. Furthermore, all ephemeral source ports observed on this test network were in the 1025-5000 range corresponding to the raised band on Figure 1b

The total traffic (DNS, Netbios, DHCP, ARP, IGMP, ICMP) generated over 99 hours by an idle private network of 5 machines (including a NAT box configured as a DHCP server) was 85 K packets (an average of one packet per 20 seconds per machine.) There were 24 K of DNS packets exchanged with our nameserver by four internal machines, i.e. on average one packet every 60 seconds. Many of the DNS queries were trying to find addresses for Active Directory names which when leaked to the Internet also go to the root servers.[17]

We only saw updates for one machine, a server without Active Directory. (We also had two Active Directory machines, and two non-server machines, Windows 2000 and Windows XP.) The RFC1918 updates directed to `prisoner.iana.org` proceed as follows (see quotes from [41] below.) The box first attempts to make an insecure UDP update by sending an update message that contains (e.g. for address 172.22.0.1)

zone: 22.172.in-addr.arpa. SOA IN
prereq: 1.0.22.172.in-addr.arpa. CNAME NONE TTL=0 1.0.22.172.in-addr.arpa.
update:1.0.22.172.in-addr.arpa. PTR ANY TTL=0 1.0.22.172.in-addr.arpa.
update: 1.0.22.172.in-addr.arpa. PTR IN TTL=1200 w2ksvrwoad.caida.rfc.

After getting `UPDATE REFUSED` in reply, it tries to make a secure update using transaction signatures (TSIG) via TCP, trying 3 times in a row. It sends 5 packets to `prisoner` on each attempt, and gets 4 packets back. This adds up to 16 packets to the server and 13 reply packets in each period. Thus for this particular Windows machine, the load that corresponds to one record in the log file amounts to 29 packets processed by `prisoner`. In our case (RTT of 80 ms) these 29 packets are exchanged in a burst that lasts about 700 ms.

Most of the inter-burst times belonged to the repeating triple of 5, 10, and 60 min interval, except two intervals that were equal 5 sec. Minimum, median, and maximum values for each interval were very close: 300.7, 300.74, 302.7 seconds respectively for 5 min interval, 600.7, 600.73, 610.1 for 10 min. and 3600.8, 3604.8, 3606.8 sec for 1 hour.

More than 14K packets were exchanged between `prisoner` and our Win2k gateway machine in 100 hours. In addition, the gateway sent three queries to our DNS server before each burst, asking for authoritative server (SOA) for its own name (configured in no-existing domain `caida.rfc`), SOA for its IP address (zone `0.22.172.in-addr.arpa`), and IP address (A) for `prisoner`.

The queries to our own DNS server were dominated address (A) for Active Directory server (`w2kwad` (extended by our fictitious domains) and about LDAP (names starting with `_ldap._tcp`) services (SRV). In addition, XP machine asked our nameserver about `_kerberos._tcp` service. This XP machine did 50% more requests than Win2k machines. Active directory Win2k machine did not do any A queries, and asked 20 times less SRV queries than other machines; in addition, this was the only machine which asked SOA queries from inside the network (NAT machine was asking them from outside.) All SOA queries for the Active Directory names were asked with average frequency of 1 per hour by both machines. This information can be used for disambiguating Windows setups.

We did not need to observe the behavior of various UNIX systems in our laboratory experiments because they typically use the BIND DNS server that requires dynamic up-

[17] About 20% of the queries at the root servers are for non-existent top level domain names like those used by Microsoft's Active Directory system [4] [10].

dates to be specifically turned on and configured with the IP address of the local primary server to be updated.

B. Microsoft documentation

The Microsoft documentation admits to both periodic update traffic and spikes at local midnight [41]. In particular, spikes at local midnight are the NETLOGON program trying to register the forward and reverse DNS mappings every 24 hours [42].[18] The 75 minute periodicity derives from the 5, 10, and 60 minute timeouts described in a Windows 2000 DNS Whitepaper [41]:

The update sequence consists of the following steps:

1. A client, using an SOA query, locates the primary DNS server and zone authoritative for the record to be registered.
2. The client sends to the located DNS server an assertion or prerequisite-only update to verify an existing registration. If the registration does not exist, the client will send the appropriate dynamic update package to register the record.
3. If the update fails the client will attempt to register the record with another primary DNS server if the authoritative zone is multimaster. If all primary DNS servers failed to process the dynamic update it will be repeated after 5 minutes and, if fails again, after another 10 minutes. If registration still failed, the described pattern of the registration attempts will be repeated after 50 minutes after the last retry.[19]

The algorithms described in the vendor's documentation treat private and non-private addreses equally. However, the non-private addresses are more likely to be allocated by a registry, in which case they must have a server authoritative for reverse mappings. The process of devolution (search for authoritative server by successive truncation of least significant parts of a domain name, used by the Windows machines [41]) will find an authoritative server at some level (historically 'class C, B, or A' address). It will not attempt the reverse record's update on the `in-addr.arpa` zone, for which root servers are authoritative. That is why Windows-generated updates for non-private adresses do not hit root servers.

VII. Conclusions and future work

RFC1918 updates and other spurious DNS traffic are a cause of a grave concern for the stability of the global Internet. As of November 2002 the number of RFC1918 update attempts at one of the major sites dedicated to processing RFC1918 requests exceeded 1300 per second, The average update rate at the same site doubled between May and October 2002. Evidence suggests that the update traffic can overwhelm routers even on most well-provisioned networks [43]. Instances of large swings in the long-term rate of update traffic are not uncommon [44].

Our preliminary analysis of possible causes of this phenomenon revealed that:

1. The bulk volume of updates surges sharply at local midnight for each time zone with a large population of Internet users. Daily/weekly update rates are consistent with common patterns of human activity.
2. The majority of updates are from sources that send them constantly. at medium sending rates.
3. Most source IP addresses are those of home and small business users connected to the Internet via cable, DSL or phone-based Internet providers. Academic, corporate and backbone networks contribute a relatively small number of updates.
4. Many update sources (close to half) use source ports in the range 1024-5000.

Our observations indicate that these illegitimate DNS updates come from computers owned by individuals, not organizations. The majority of them use the software with default vendor settings. It is natural to assume, especially in light of observation (2), that persistent update generation is the default behavior of Microsoft's DNS implementation. To find out precisely what causes the periodic RFC1918 update traffic, we took the following steps:

- attemped OS fingerprinting using publicly available utility Xprobe, with inconclusive results.
- analyzed interarrival times for aggregate traffic and per source, identifying two narrow spikes in the per-source frequency spectrum with a specially tailored autocorrelation function. The respective periods were found to reflect one update per hour and 3 updates per 75 minutes.
- set up laboratory experiment with off-the-shelf software confirming that Windows (2000 and XP) DHCP/DNS servers send periodic DNS updates and use the above mentioned port range.
- found Microsoft documentation describing their DNS update implementation with observed periods as the default behavior for their operating systems.

Prior to the deployment of the AS112 authoritative servers for RFC1918 address space (Spring 2002), Microsoft desktop machines with private addresses were attempting to update the DNS root servers who are authoritative for the `in-addr.arpa` top level domain. That can be compared to a slowly paced, massive, distributed denial of service (DDoS) attack on the root name server system.

We have demonstrated that the vast majority of periodic update behavior derives from two specific operating systems: Windows 2000 and Windows XP. We conclude that Microsoft must change the default configuration of Windows systems so that dynamic DNS updates are disabled by default, and so that user configuration, or lack thereof, does not enable RFC1918-related traffic to propagate beyond the local subnet.

More generally we consider this study a compelling example of why software and setups affecting stability of the Internet's infrastructure must be designed with more careful attention to potential effects of engineering deci-

[18] "By default, DNS records are re-registered dynamically and periodically every 24 hours by Windows 2000 Professional and every 1 hour by Windows 2000 Server and Windows 2000 Advanced Server." [42] "A statically configured client does not communicate with the DHCP server and dynamically updates A and PTR RRs every time it boots up, changes its IP address or per-adapter domain name" [41]

[19] This is probably a typo: our laboratory measurements revealed a delay of 60 minutes, not 50 minutes.

sions/misimplementations on global systemic Internet stability.[20] Indeed the current state of desktop software poses a substantial and increasing burden on, if not threat to, the robustness of the global Internet.

A. Future work

Our results show that dynamic DNS updates contain a wealth of information about networks at the Internet's edge that cannot be gleaned by other means. For example, they may be used for to assess trends in deployment of personal computers and home networks as reflected by their traffic to anycast name servers, for clock drift estimation of a typical consumer PC, and other macroscopic questions for which no reliable method is currently available.

In the future it may also be possible to develop techniques of OS fingerprinting via update sequences. It should also be possible to use robust spectral analysis of DNS updates (such as that presented here) to estimate how many end systems use DHCP and NAT.

In October 2002 - June 2003 we collected a continuous set of logs that incudes data from the topological proximity of two root servers. This data can shed light on the long-term feasibility of anycast routing, its interaction with the changes propagated by BGP, and effects of floods (e.g., worms) on end user connections.

Yet another interesting question is how many superfluous queries that reach root servers [10] are leaked by Microsoft-based networks, and how subsequent Windows releases will alleviate or exacerbate these problems.

B. Acknowledgements

Many thanks to Paul Vixie and Peter Losher of Internet Software Consortium, Brian Kantor of UCSD, Piet Barber of Verisign, Cricket Liu of Men and Mice, Tom Guptill of SDSC, and to our CAIDA colleagues Nevil Brownlee, Grant DuVall, Brad Huffaker, Dan Andersen, Marina Fomenkov, Ken Keys, Young Hyun. Availability of Xprobe by Ofir Arkin of Sys-Security Group is gratefully acknowledged. Thanks to IPAM UCLA Large-Scale Communication Networks program (March-June 2002) where our approach of network spectroscopy was formulated. The feedback from the IETF participants and NANOG mailing list subscribers for the early drafts of this paper, and from SIGMETRICS and WIAPP reviewers was also highly appreciated.

References

[1] Paul Ablitz and Cricket Liu, "DNS and BIND," 2001, O'Reilly.
[2] Cricket Liu, "DNS and BIND Cookbook," 2002, O'Reilly.
[3] P.Mockapetris, "Domain names - implementation and specification, RFC1035," November 1987.
[4] Nevil Brownlee, kc claffy, and Evi Nemeth, "DNS Measurements at a Root Server," Globecom 2001.
[5] Stuart Cheshire, Bernard Aboba, and Erik Guttman, "Dynamic Configuration of IPv4 Link-Local Addresses," draft-ietf-zeroconf-ipv4-linklocal-07.txt, 23rd August 2002 http://www.potaroo.net/ietf/ids/draft-ietf-zeroconf-ipv4-linklocal-07.txt.

[20] As an example, quality assurance for network software should include measurement of traffic leaking to root nameservers (or other shared global resources) in a variety of configurations.

[6] Nevil Brownlee, kc claffy, and Evi Nemeth, "DNS Root/gTLD Performance Measurements," Usenix LISA, 2001.
[7] Marina Fomenkov, kc claffy, Bradley Huffaker, and David Moore, "Macroscopic Internet Topology and Performance Measurements From the DNS Root Name Servers," Usenix LISA, 2001.
[8] A. Broido, E. Nemeth, and kc claffy, "Interdomain Routing Evolution - Episode 2: Dark Space," presented at ARIN IX meeting, Las Vegas, April 02, 2002, www.caida.org/outreach/presentations/arin0402/.
[9] Duane Wessels, "Toward lowering the load on root DNS servers," Presented at the NANOG meeting, Eugene, Oct.27-30, 2002.
[10] Duane Wessels and Marina Fomenkov, "Wow, that's a lot of packets," in *Proceedings of Passive and Active Measurement Workshop (PAM), San Diego*, April 2003.
[11] A. Broido, E. Nemeth, and kc claffy, "Spectroscopy of DNS update traffic," in *ACM SIGMETRICS*, June 2003.
[12] Richard Liston, Sridhar Srinivasan, and Ellen Zegura, "Diversity in dns performance measures," in *Proceedings of the IMW*, Nov 2002.
[13] A. Broido, R. King, E. Nemeth, and kc claffy, "Radon spectroscopy of inter-packet delay," in *Proceedings of the IEEE High-Speed Networking Workshop, San Francisco, March 2003.*
[14] Tim Griffin, "What is the sound of one route flapping?," www.cs.dartmouth.edu:80/23.pdf.
[15] Olaf Maennel and Anja Feldmann, "Realistic BGP Traffic for Test Labs," in *ACM SIGCOMM 2002, Pittsburg, PA*, Aug 2002.
[16] David Andersen, Nick Feamster, Steve Bauer, and Hari Balakrishnan, "Topology inference from bgp routing dynamics," in *Proceedings of the IMW*, Nov 2002.
[17] Dina Katabi and Charles Blake, "Inferring congestion sharing and link characteristics from packet interarrival times," MIT LCS Technical Report, 2001.
[18] Mark Coates, Alfred Hero, Robert Nowak, and Bin Yu, "Internet tomography," vol. 19, May 2002.
[19] Ravi S. Prasad, Constantinos Dovrolis, and Bruce A. Mah, "The effect of layer-2 switches on pathchar-like tools," in *Proceedings of the IMW*, Nov 2002.
[20] Mark Crovella, Anukool Lakhina, John Byers, and Ibrahim Matta, "Where on earth is the internet?," Dec 2001, ISMA workshop 2001. San Diego, CA, `www.caida.org/outreach/isma/0112/talks/mark/`.
[21] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear, "Address Allocation for Private Internets, RFC1918," February 1996.
[22] Tony Bates, "Classless Inter-Domain Routing (CIDR) Report," http://www.employees.org/ tbates/cidr-report.html.
[23] A. Broido, E. Nemeth, and kc claffy, "Fringe Address Spaces," in preparation.
[24] P. Srisuresh and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT), RFC3022," January 2001.
[25] R.Droms, "Dynamic host configuration protocol, RFC2131," March 1997.
[26] S.Alexander and R.Droms, "DHCP options and BOOTP vendor extensions, RFC2132," March 1997.
[27] P.Vixie, S.Thompson, Y.Rekhter, and J.Bound, "Dynamic updates in the Domain Name System (DNS updates), RFC2136," April 1997.
[28] "AS112 Project Home Page," www.as112.net.
[29] Paul Vixie, "Is your host or DHCP server sending DNS dynamic updates for RFC1918?," NANOG mailing list, April 2002, www.irbs.net/internet/nanog/0204/0450.html.
[30] "Root Server Technical Operations Association," www.root-servers.org.
[31] Brian Kantor, ," private communication July 4, 2002.
[32] Bill Manning, "Documenting Special Use IPv4 Address Blocks that have been registered with IANA," draft-manning-dsua-04.txt, 03 Jan 2001, www.isi.edu/ bmanning/dsua.html.
[33] "SBC Yahoo DSL Pricing," www.pacbell.com.
[34] D. Moore, R. Periakaruppan, J. Donohoe, and k claffy, "Where in the World is netgeo.caida.org?," Proceedings of Inet '2000, `http://www.caida.org/outreach/papers/inet_netgeo/`.
[35] David Meyer, "University of Oregon Route Views Archive Project, www.routeviews.org," .
[36] Nevil Brownlee and kc claffy, "Understanding Internet Traffic Streams: Dragonflies and Tortoises," IEEE Communications.
[37] Y. Zhang, V.Paxson, and S.Shenker, "The Stationarity of Inter-

net Path Properties: Routing, Loss and Throughput," in *ACIRI Technical Report*, May 2000.
[38] Steven M. Bellovin, "A Technique for Counting NATted Hosts," in *Proceedings of IMW*, 2002.
[39] V. Paxson and S. Floyd, "Wide-area traffic: The failure of poisson modeling," Jun 1995.
[40] Ofir Arkin, "Fingerprinting utility Xprobe 1," www.sys-security.com.
[41] "Windows 2000 DNS Whitepaper," www.microsoft.com/windows2000/docs/w2kdns.doc.
[42] "How to Enable/Disable Windows 2000 Dynamic DNS Registrations," Microsoft Knowledge Base Article - Q246804.
[43] Piet Barber, ," September 11 2002, e-mail to the authors.
[44] Akira Kato, ," February 21 2003, e-mail to the authors.