



Recent Authoritative Exhaustion Attacks

Chris Baker

DNS OARC 2016 Dallas

**INTERNET
PERFORMANCE.
DELIVERED.**

 dyn.com  [@dyn](https://twitter.com/dyn)

<https://github.com/jgamblin/Mirai-Source-Code/blob/master/mirai/prompt.txt>

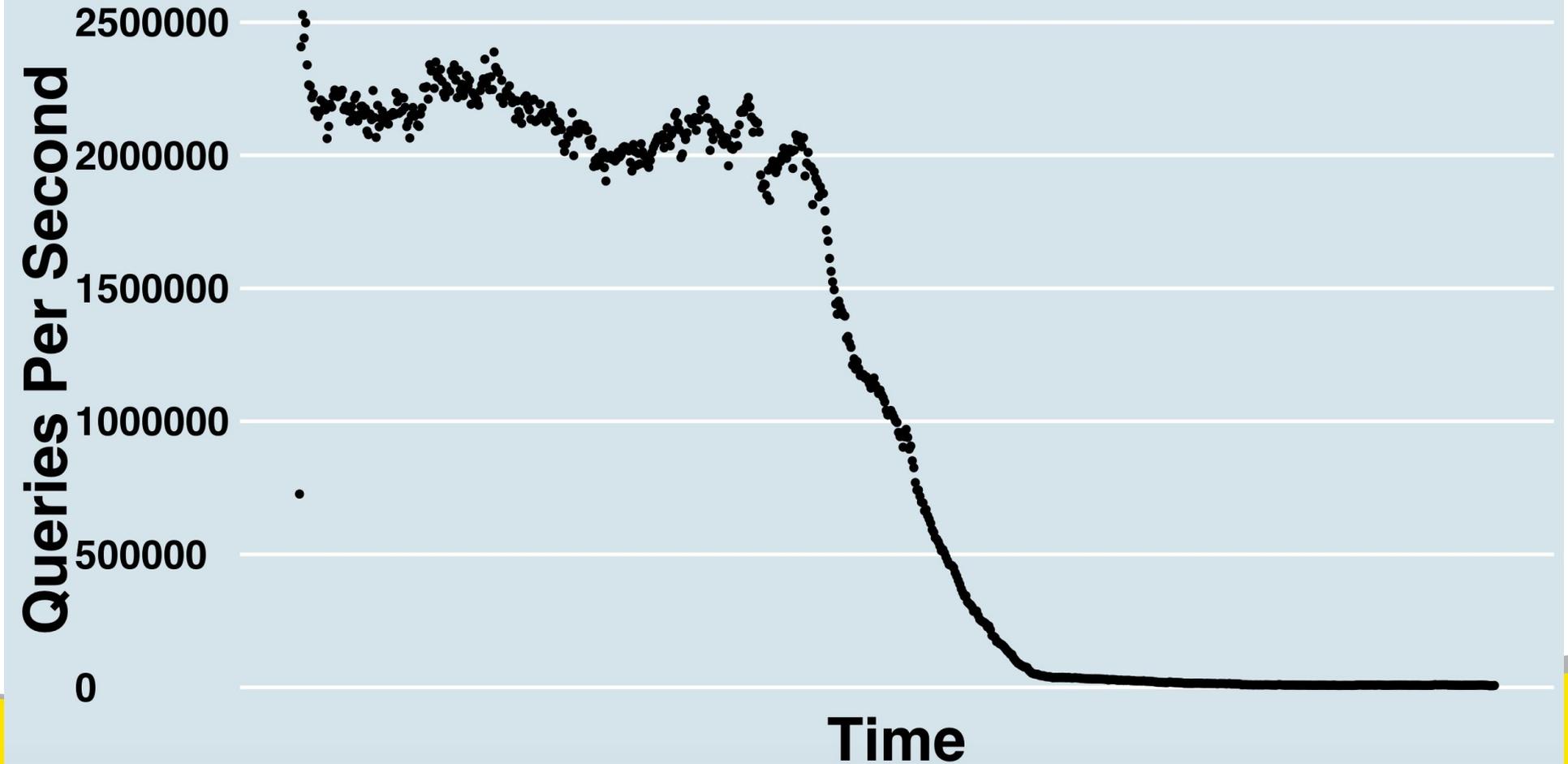
Introduction

- In early August we started to see an uptick in DDoS attacks
 - Both targeting our customers and observed in recursive traffic
- These were out of the norm when contrasted with traditional booter / stresser attack payloads
 - Not DNS amplification using cpssc.gov, SSDP, NTP, Chargen ... etc
- The attacks fit the profile of authoritative DNS exhaustion
 - A large volume of properly formed in protocol queries
 - Targeting domains that were delegated to Dyn's nameservers
 - Recursive cache busting set of pseudo-random characters
 - Example: lq18v2V3N2lQ.<sub domain>.<domain>.<tld>
 - The 12 character pseudorandom string attached to the valid domain was a consistent attribute
 - "Random" seems to exclude certain values 'z' and '9' for example

How scary does this look?

- Recent trends in attack patterns have lead operators to assume that the packets sources are spoofed until proven authentic
- One thing that stood out when looking was the traffic was the distribution of qnames with 0x20 bit character randomization.
- Not only did it stand out but, it matched known patterns
 - Traffic from resolvers known to have implemented 0x20 bit randomization were consistently randomized
 - At that point it became reasonable to assume that this traffic was being generated by clients and passing through the recursive layer

Queries Per Second Timeseries



Into the Recursive Layer!

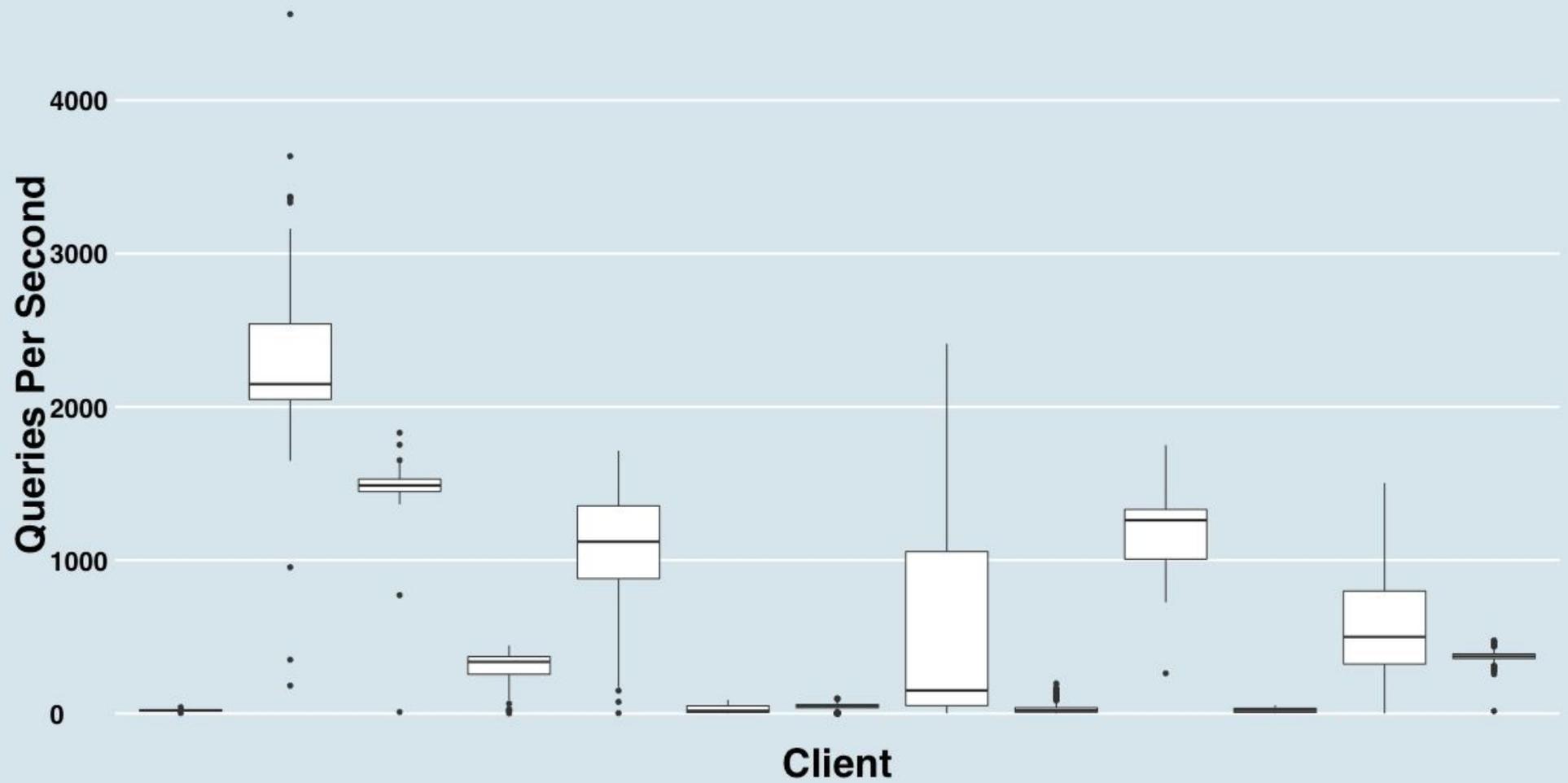
Dyn operates a recursive resolver platform

A handful of infected devices were configured to use Dyn's recursive resolvers from an array of different autonomous systems

This provides us with some data around the number of queries per second individual client IPs were issuing

The variability in the rate of queries per second and the IPs which were involved in one attack and not other was also of interest

Queries Per Second by End Point



What are they?

Apache2 Ubuntu Default Page x http://2.178.192

49.156.151



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed correctly. You should **replace this file** (located at `/var/www/html/index.html`) on your HTTP server.

If you are seeing this web site and don't know what this page is about, this probably means the site is unavailable due to maintenance. If the problem persists, please contact the system administrator.

Configuration Overview

The default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

62.119.183

It works!

Country Code	Number of Prefixes	Count of ASNs	% of Traffic
US	226	99	31.54
CN	269	42	15.12
BR	580	264	8.33
RU	522	373	5.58
TW	22	12	3.88
UA	181	139	3.03
KR	26	14	2.95
BG	128	86	2.71
CL	18	9	1.95
ID	35	21	1.94

Attack Source Code Leaked on HackForums

```
void attack_udp_dns(uint8_t targs_len, struct attack_target *targs, uint8_t opts_len, struct attack_option *opts)
{
    int i, fd;
    char **pkts = calloc(targs_len, sizeof(char *));
    uint8_t ip_tos = attack_get_opt_int(opts_len, opts, ATK_OPT_IP_TOS, 0);
    uint16_t ip_ident = attack_get_opt_int(opts_len, opts, ATK_OPT_IP_IDENT, 0xffff);
    uint8_t ip_ttl = attack_get_opt_int(opts_len, opts, ATK_OPT_IP_TTL, 64);
    BOOL dont_frag = attack_get_opt_int(opts_len, opts, ATK_OPT_IP_DF, FALSE);
    port_t sport = attack_get_opt_int(opts_len, opts, ATK_OPT_SPORT, 0xffff);
    port_t dport = attack_get_opt_int(opts_len, opts, ATK_OPT_DPORT, 53);
    uint16_t dns_hdr_id = attack_get_opt_int(opts_len, opts, ATK_OPT_DNS_HDR_ID, 0xffff);
    uint8_t data_len = attack_get_opt_int(opts_len, opts, ATK_OPT_PAYLOAD_SIZE, 12);
    char *domain = attack_get_opt_str(opts_len, opts, ATK_OPT_DOMAIN, NULL);
    int domain_len;
    ipv4_t dns_resolver = get_dns_resolver();

    if (domain == NULL)
    {
#ifdef DEBUG
        printf("Cannot send DNS flood without a domain\n");
#endif
    }
}
```

If Local Resolver Not Found ...

```
switch (rand_next() % 4)
{
case 0:
    return INET_ADDR(8,8,8,8);
case 1:
    return INET_ADDR(74,82,42,42);
case 2:
    return INET_ADDR(64,6,64,6);
case 3:
    return INET_ADDR(4,2,2,2);
}
```

This explains why 34% of the traffic is being attributed to the US

Google
Hurricane Electric
Verisign
Level 3

The defaults make up 24% of aggregate traffic.

Duration

The DNS attacks from Mirai targeting our customers lasted for about 10 mins.

Some researchers have attributed this to the limitations of the command and control infrastructure relative to the size of the infrastructure being managed

Other researchers attribute this to issues with the stability of the devices the attack is being launched from.

Anna-Senpai, the handle which dumped the source code, mentioned the botnet was ~380K devices however post Krebs DDoS the number was down to 300K and dropping due to ISP reactions

Other Patterns Emerging

We have observed another / other bot(s) with authoritative exhaustion functionality

- The character set being used is different

 - Only lowercase letters - abcdefghijklmnopqrstuvwxyz

- The length of the cachebuster varies from 4 - 16

- This bot is able to keep long running attacks

From samples obtained, this can't be tied back to any known qbot / lizkebab variants

A globe is shown on the left side of the slide, partially obscured by a complex network of white lines and nodes that represent a global network or data flow. The background is black, and the text is in a bright yellow color.

Thank You!