WHITE PAPER

Debugging DNS

NS1.

Introduction

DNS is mission-critical to all businesses that connect to the Internet. If you are moving part or all of your business to the Cloud that reality becomes even more apparent. In order to resolve DNS issues that arise on your network, IT professionals need to understand DNS, industry best practices, and available tools to help you resolve these issues before they cost your business time and money.

This paper provides an overview of the Domain Name Service(DNS) along with best practices to keep your network running smoothly and suggestions to resolve problems when it does not.

How DNS Name Resolution Works

The key to how the Internet works is the distributed system of name resolution. The larger the Internet becomes; the more vital name resolution is. DNS links the human readable domain name to an identification number in either IPv4 or IPv6 format.

Domain names are read left to right, but they are resolved right to left starting with the final dot, usually assumed and not entered into a browser address. Here is an example of a Fully Qualified Domain Name and how DNS will find its IPv4 or IPv6 addresses, assuming the name resolution is not cached anywhere.

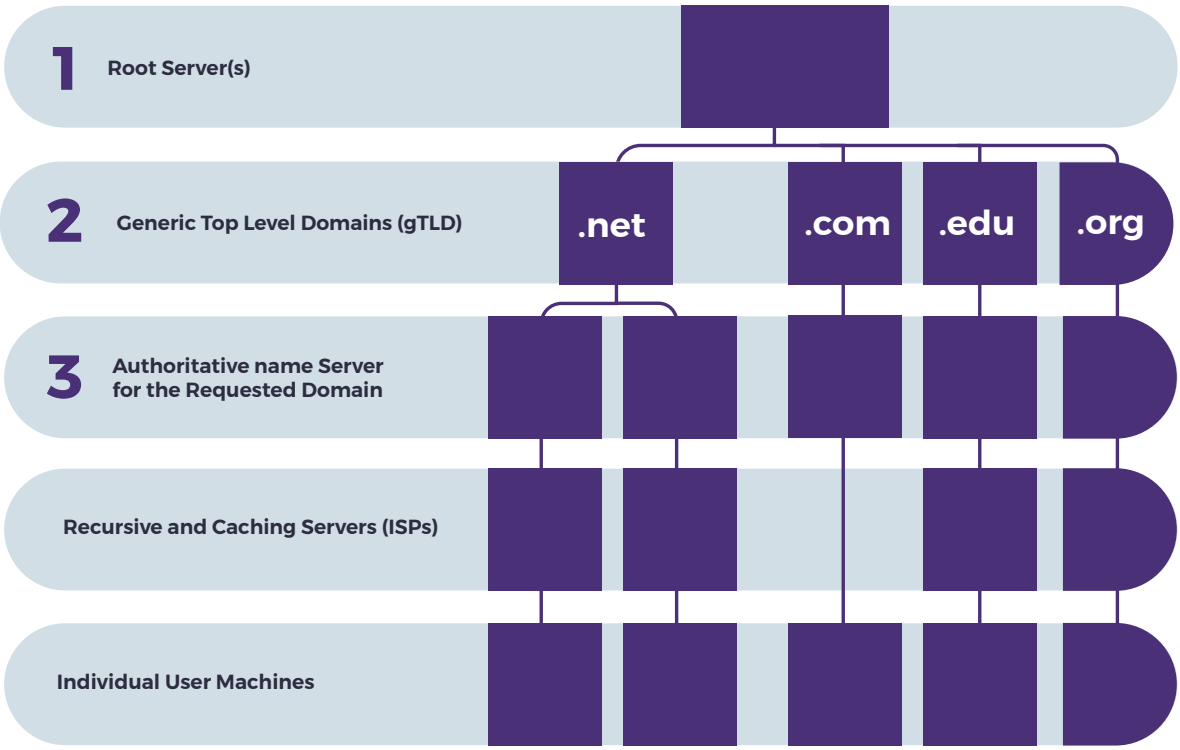
www	.	example	.	com	.
4 Sub-domain name label	separator	3 Domain name label	separator	2 gTLD label	1 root label

1. Root label is the assumed dot at the end of every domain name. It signifies the root of the DNS. It is normally not included when writing domain names, but is always considered part of the fully qualified domain name.

2. gTLD label indicates the generic Top Level Domains of the Internet.

3. Domain Name label is the human readable name of the domain being queried.

4. Sub-domain name label(s).



To resolve www.example.com, the Recursive DNS Resolver will iteratively query the authoritative nameservers for each level of recursion. First it will query the root servers (1) for the name servers for the gTLD of the target domain, .com in this case. It will then query those gTLD servers (2) for the nameservers of the target domain, (3) example.com, Next it will query the target domain's nameservers for the A record for (4) www.example.com.

Each time a DNS resolver receives an answer to a query, it caches that answer for future use. The answer has a limited time, called the time to live (TTL), before it needs to be refreshed from the authoritative name server. Because each system caches the answers it receives, a DNS query to a common location may be resolved at the user's ISP resolver or even on the user's own computer.

Recursive and Caching servers are often hosted by ISPs for their customers. This allows DNS name resolution to occur closer to the end users.

Tools To Help Identify DNS Issues

The following tools are available online or for download. Some may be included in your operating system. They are provided here as a convenience and not as an endorsement.

Dig
An open-source DNS diagnostic tool. Dig is the standard and preferred tool to diagnose DNS issues throughout the industry.www.digwebinterface.com, toolbox.googleapps.com/apps/dig/. Dig +trace is a command for the Dig tool that traces a DNS query's route from the DNS root servers to the requested server.

DNS Map / What's My DNS
Online tools to determine how DNS resolves a domain name without using the local ma-chine's resolver.

NSlookup
Stock tool available on all operating systems to perform basic DNS diagnostics.

Traceroute
A utility that records the route through the Internet between your computer and a specified destination. It also calculates and displays the amount of time each hop along the route took to complete. This tool comes installed on many operating systems, or you can download the utility from several websites.

Whois
An online service available from several websites (www.whois.net, whois.domaintools.com, whois.icann.org, are some examples) that identifies the registered owner of a given domain name. The information provided also includes contact information for the owner, domain status, when the registration expires, and the name servers for the domain.

Because domain ownership information is publicly available, some domain owners prefer to purchase a secret registration where the displayed name, address, and contact information in the whois record belong to the registration company and the registration company maintains the domain ownership records separate from the publicly available whois record.

Debugging / Troubleshooting Example

This example walks through the common scenario of a recursive server that returns the wrong DNS record. Usually the reason for this is that the recursive server has cached an old DNS record. Cached records are updated when the record's time to live (TTL) expires.

1. Customer identifies that they cannot reach your website, example.com.
   NOTE: The nameserver domains in this example have been replaced with the domain name exampledns. When you run dig, you will see the names of the actual recursive or authoritative domains in the fifth column of the results.
2. Run dig example.com and view the results. This will show you the DNS records that are cached at the recursive server.

```
$ dig example.com
; <<>> DiG 9.9.5-3ubuntu0.11-Ubuntu <<>> example.com
;; global options: +cmd
;; Got answer:
;; ->> HEADER<<- opcode: QUERY, status: NOERROR, id 60335
;; flags: qr rd ra; QUERY: 1, ANSWER 6, AUTHORITY: 6, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;example.com. IN A
;; ANSWER SECTION:

example.com. 60 IN A 54.239.25.208
example.com. 60 IN A 54.239.17.6
example.com. 60 IN A 192.45.182.45
example.com. 60 IN A 54.239.26.128
example.com. 60 IN A 264.45.52.5
example.com. 60 IN A 54.239.17.7

;; AUTHORITY SECTION:
example.com. 172800 IN NS pdns6.exampledns.co.uk.
example.com. 172800 IN NS pdns1.exampledns.net.
example.com. 172800 IN NS ns1.p31.exampledns.net.
example.com. 172800 IN NS ns4.p31.exampledns.net.
example.com. 172800 IN NS ns3.p31.exampledns.net.
example.com. 172800 IN NS ns2.p32.exampledns.net.

;; Query time 186 msec
;; SERVER 127.0.0.1#53(127.0.0.1)
;; WHEN: Tue Mar 01 12:01:12 EST 2017
;; MSG SIZE rcvd: 284
```

3. To verify the results from dig, run dig example.com +trace.
This will show you the DNS records at the authoritative nameservers.

```
$ dig example.com +trace
; <<>> Dig 9.9.5-3 ubuntu0.11-Ubuntu <<>> example.com +trace
;; global options +cmd

. 439081 IN NS h.root-servers.net.
. 439081 IN NS i.root-servers.net.
. 439081 IN NS f.root-servers.net.
. 439081 IN NS m.root-servers.net.
. 439081 IN NS k.root-servers.net.
. 439081 IN NS e.root-servers.net.
. 439081 IN NS b.root-servers.net.
. 439081 IN NS g.root-servers.net.
. 439081 IN NS d.root-servers.net.
. 439081 IN NS a.root-servers.net.
. 439081 IN NS c.root-servers.net.
. 439081 IN NS j.root-servers.net.
. 439081 IN NS l.root-servers.net.
. 515662 IN RRSIG NS 8 0 518400
20170314150000 20170301140000 61045 . bEGgigAVufDUAwjq9p0fiW/OU+cvicFY-
OVqKv5Tb6PsZEVdG7KCf/1PtugjsWdjsouyinAtvjKylVMLBANa04fUbu43VW7AML2vrUxQ9t7YgB
M5+1dc80DPq0w4880MF9pEhr20AhyEvamxG1qjfAUGDXU7rqs66Rfz0VCQ45AzT7Ne4T31KSGQ/
kLyInQeyfjEAtzqJLgg9IODa8v9KhS+wTabqa6TH5hDwXkgoPTMKf6uoVvneR VgMmZyl0YJ1gwHd6CMnPnt-
JZs600rcp4JZjWmoRIOEFUGcAZLmmrdMYA/cnMZQ==

;; Received 1097 bytes from 127.0.0.1#53(127.0.0.1) in 8ms
com. 172800 IN NS h.gtld-servers.net.
com. 172800 IN NS i.gtld-servers.net.
com. 172800 IN NS f.gtld-servers.net.
com. 172800 IN NS m.gtld-servers.net.
com. 172800 IN NS k.gtld-servers.net.
com. 172800 IN NS e.gtld-servers.net.
com. 172800 IN NS b.gtld-servers.net.
com. 172800 IN NS g.gtld-servers.net.
com. 172800 IN NS d.gtld-servers.net.
com. 172800 IN NS a.gtld-servers.net.
com. 172800 IN NS c.gtld-servers.net.
com. 172800 IN NS j.gtld-servers.net.
com. 172800 IN NS l.gtld-servers.net.
com. 86400 IN DS 30909 8 2EZD3C916F6DEEAC73294E8268FB5885044A833FC459588F4A9184CFC41A5766
com. 86400 IN RRSIG DS 8 1 86400
20170314150000 20170301140000 61045 . bEGgigAVufDUAwjq9p0fiW/OU+cvicFY-
OVqKv5Tb6PsZEVdG7KCf/1PtugjsWdjsouyinAtvjKylVMLBANa04fUbu43VW7AML2vrUxQ9t7YgB-
M5+1dc80DPq0w4880MF9pEhr20AhyEvamxG1qjfAUGDXU7rqs66Rfz0VCQ45AzT7Ne4T31KSGQ/
kLyInQeyfjEAtzqJLgg9IODa8v9KhS+wTabqa6TH5hDwXkgoPTMKf6uoVvneR  VgMmZyl0YJ1gwHd6CMnPnt-
JZs600rcp4JZjWmoRIOEFUGcAZLmmrdMYA/cnMZQ==
```

```
;; Received 1097 bytes from 127.0.0.1#53(127.0.0.1) in 8ms

example.com. 172800 IN NS pdns6.exampledns.co.uk.
example.com. 172800 IN NS pdns1.exampledns.net.
example.com. 172800 IN NS ns1.p31.exampledns.net.
example.com. 172800 IN NS ns4.p31.exampledns.net.
example.com. 172800 IN NS ns3.p31.exampledns.net.
example.com. 172800 IN NS ns2.p32.exampledns.net.
CK0POJMG874ljref7efn8430QVIT8BSM.com. 86400 IN NSEC3 1 1 0 -
CK0Q1GIN43N1ARRC9OSM6QPQR81H5M9A NS SOA RRSIG DNSKEY NSEC3PARAM

CK0POJMG874ljref7efn8430QVIT8BSM.com. 86400 IN RRSIG NSEC3 8 2
86400 20170306054755

20170227043755 31697 com. HR4ZLV6E01sGvY024zq7qQvat3rZABR7Cmb3u0o
fGXPVUratFwqpln8cQR+kFg1+n3k09NCL88tsd6MIMFkjkOH2GP5RKWI
qFvNXq18snt5rXHWhlCsdIbhxa4Leu7jqSrPF6Si01gQM8/+L6RZv7Wx RKc=

K1VUKD24902LTRO6Q8TIJ34K9HT9D2G2.com. 86400 IN NSEC3 1 1 0 -
K201UOPSV9V35UT2FNV6I40LA898RML1 NS DS RRSIG

K1VUKD24902LTRO6Q8TIJ34K9HT9D2G2.com. 86400 IN RRSIG NSEC3 8 2
86400 20170307054953 20170228043953 31697 com. fGXPVUratFwqpln8cQR+kFg1+n3k09NCL88tsd-
6MIMFkjkOH2GP5RKWI qFvNXq18snt5rXHWhlCsdIbhxa4Leu7jqSrPF6Si01gQM8/+L6RZv7Wx k3A=

;; Received 781 bytes from 192.52.178.30#53(k.gtld-servers.net) in 200ms
example.com. 60 IN A 54.239.25.208
example.com. 60 IN A 54.239.17.6
example.com. 60 IN A 54.239.25.192
example.com. 60 IN A 54.239.26.128
example.com. 60 IN A 54.239.25.200
example.com. 60 IN A 54.239.17.7
example.com. 172800 IN NS pdns6.exampledns.co.uk.
example.com. 172800 IN NS pdns1.exampledns.net.
example.com. 172800 IN NS ns1.p31.exampledns.net.
example.com. 172800 IN NS ns4.p31.exampledns.net.
example.com. 172800 IN NS ns3.p31.exampledns.net.
example.com. 172800 IN NS ns2.p32.exampledns.net.

;; Received 284 bytes from 204.13.251.31#53(ns4.p31.exampledns.net.) in 7ms
```

4. Compare the results from the recursive server and the authoritative nameservers. Note that the underlined A records have different IP addresses at the authoritative server than they did at the recursive server in step 2. The authoritative servers are considered the most correct information.

5. If there is a discrepancy, it is most likely that the recursive server needs to refresh its records. The second column of each dig / dig+trace response shows the TTL in seconds. Wait the time designated by the TTL and re-run dig and dig+trace to confirm that the recursive server's information has been updated to match the authoritative server. In most cases, this will resolve the name resolution issues seen by the customer.

Troubleshooting Checklist

Here is a general checklist for troubleshooting domain name resolution issues.

1. Find the nameserver for your domain.
   Examples: `nslookup, whois, dig`
2. Verify the A record by querying the nameserver directly to bypass caching.
   Examples: `dig`
3. If the answer you receive from your recursive DNS server is different from the answer from the Authoritative Nameservers, the old answer is likely cached, or there may be an issue with advanced features with your DNS provider.

   a. The ISP's intermediary cache (checking only, generally)
   Example: `dig`

   b. Local cache
   Examples: `ipconfig (Windows), mDNSresponder (OS X), /etc/init.d/nscd (Linux)`

   c. Browser cache
   Example: `chrome://net-internals/#dns (Chrome)`
4. Compare IPs and check TTL
   Examples: `dig`
5. Double-check any updated records.
   Examples: `dig`
6. Verify visibility.
   Examples: `nslookup`

DNS Best Practices

Prepare TTLs for Planned Changes: When updating a DNS record, first reduce the TTL to a very small value and wait an amount of time equal to the original TTL value before actually changing the record. Once you've tested and confirmed the new setup, change the TTL back. Managing the TTL with this method allows the old record to expire in any caches, significantly reducing the window in which traffic can be accidentally directed to the wrong location.

Monitor continuously: Many Managed DNS providers include monitoring software with their services. Because DNS issues may not be apparent until they become significant, using monitoring software to alert you when an unusual situation arises allows you to catch DNS issues on your network early.

Design redundancy into the network: Server drives fail, switches need replacing, any number of hardware or software issues could happen to your network. Failures due to age, overload, or acts of nature happen. By designing redundancy into all aspects of your network, including your DNS system, you avoid having any single item in your network bringing the entire network down.

Use Geographically diverse DNS servers: Natural disasters happen all over the world. By diversifying the physical locations where you place your DNS servers you reduce the probability that your network will be taken offline by any single natural disaster.

Secure your system: There is physical security for your system hardware and there is online security for your network.

- Allow transfers only from trusted servers. This reduces the probability that a hacker will gain access to your network resources.
- Disable recursion on authoritative servers. Disabling recursion on authoritative servers lowers the possibility that the server can be used in a DDoS attack.
- Use a hidden master DNS server. A master DNS server, also called a primary DNS serv-er, is one that is authoritative for its included zones. Hiding the master server removes it from the list of NS records on the domain. Other servers, secondary or other masters, can meet the need of having at least 2 NS records per domain.

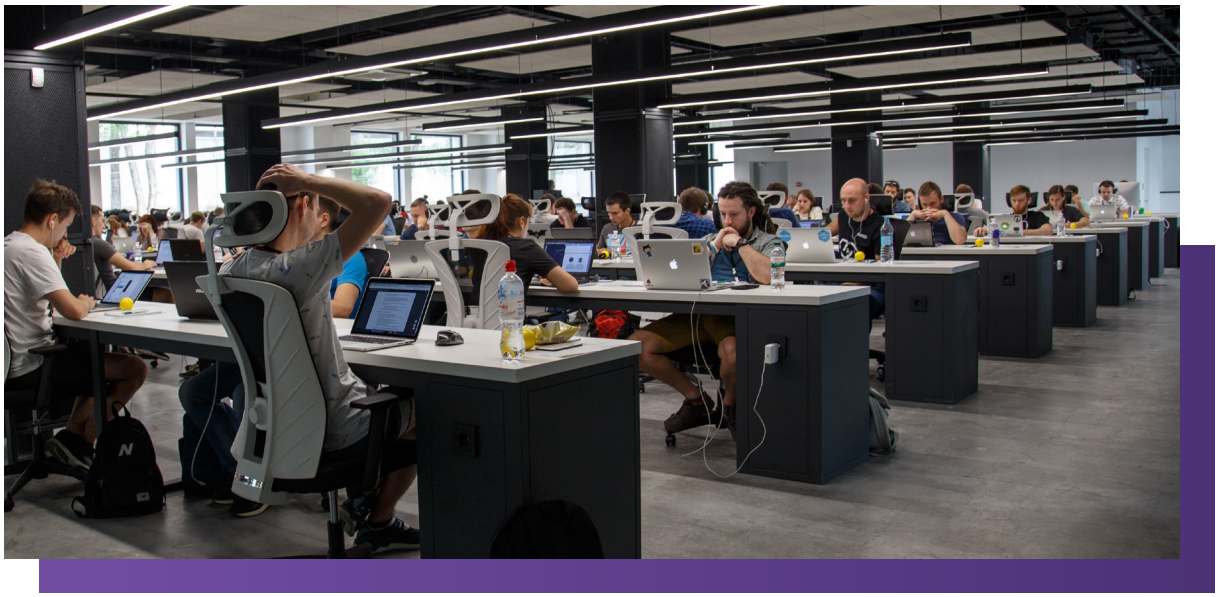
NS1 offers enterprise DNS solutions for your private network and internet facing online services.

Managed DNS
Cloud based, intelligent DNS for internet facing online services

Private DNS
NS1's carrier grade DNS platform for self-hosted deployments

Dedicated DNS
A single tenant managed solution for DNS redundancy

Pulsar
Real user measurement based traffic routing for application optimization



About NS1

NS1 is the leader in next generation DNS solutions that orchestrate the delivery of the world's most critical internet and enterprise applications. Only NS1's purpose-built platform, which is built on a modern API-first architecture, transforms DNS into an intelligent, efficient and automated system, driving dramatic gains in reliability, resiliency, security and performance of application delivery infrastructure. Many of the highest-trafficked sites and largest global enterprises trust NS1, including Salesforce, LinkedIn, Dropbox, Nielsen, Squarespace, Pandora and The Guardian.