DOMAIN NAME SYSTEM

OVERVIEW

The domain name system (DNS) is a distributed, hierarchical data store and name resolution service that is used primarily to resolve IP addresses based on much more user-friendly domain names. DNS is a powerful and rather complex system that plays a variety of crucial roles in modern systems and applications. It is so central to these systems that any degradation of service will severely impact the usability of a network-connected computer or mobile device (e.g., resulting in extremely slow boot and application launch times).

Our objective is to gain a functional understanding of the system, which we can apply broadly to our work in other areas of the informatics domain. The following videos provide a highly accessible starting point.

- Pieter Explains Tech https://www.youtube.com/watch?v=GlZC4Jwf3xQ
- Khan Academy https://www.khanacademy.org/computing/computer-science/internet-intro/internet-works-intro/v/the-internet-ip-addresses-and-dns

DNS NAMESPACE AND ZONES

As you've seen, domain names are structured in a hierarchical manner that reflects the decentralized nature of the domain name system. The DNS namespace can be represented by a tree, with branches descending from a common root. A fully-qualified domain name (FQDN) is written by concatenating labels from left-to-right, tracing the path from a leaf node to the root by way of the parent nodes. A trailing dot, representative of the root, is appended to the FQDN for DNS operations though it is omitted in most user-facing applications.

Below the root, the top-most level of the graph represents top-level domains (TLD), such as com, net, and edu, that appear at the end of every FQDN. IANA and its parent organization ICANN oversee the governance of the DNS root and top-level domains, though it delegates registration functions and operation of root and TLD name servers to independent organizations.

Names appearing as children of each TLD are assigned to organizations through the domain registration process. An organization registers a domain name by working with an ICANN-accredited registrar or a reseller subcontracted by an accredited registrar.

- https://whois.icann.org/en/domain-name-registration-process

Further subdivisions of the registered namespace can be created by the registrant. These subdomains appear as children of the registrant domains in the DNS tree described previously.

DNS ZONES

The hierarchical structure of DNS lends to dividing administrative responsibility of the global namespace. The term "zone" refers to these administrative divisions. Each zone in DNS is a contiguous portion of the namespace under the administrative responsibility of a single manager. While zones are often aligned with individual domain registrations, the relationship is not one-to-one. A zone may include multiple domains. Likewise, a domain may be divided into additional zones encapsulating distinct subtrees.

The root zone database is managed by IANA and contains the authority records for the TLDs. Likewise, IANA assigns operators for the TLDs. These operators maintain the authoritative database of all domain names and authority records registered in the TLD. These databases are referred to as the DNS registries and can be queried to identify the authoritative nameservers for child zones.

DNS NAME SERVERS

Each zone, including the root, is hosted by one or more name servers that store collection of resource records associated with the corresponding subset of the DNS namespace. As mentioned previously, the root name servers host authority records for each TLD registry. Likewise, TLD name servers host authority records for child zones.

- https://www.lifewire.com/dns-root-name-servers-3971336
- https://www.iana.org/domains/root/servers
- http://root-servers.org

Name servers come in several varieties. In addition to the root servers and TLD servers described above, resource records for each zone are hosted by at least one authoritative name server. These authoritative name servers may be hosted by the registrant itself or by a third-party DNS hosting service. Many organizations maintain internal, private zones serving records that are only available within well-defined network boundaries.

Not all name servers are authoritative for a zone. Plenty of servers are created to satisfy service requirements for performance, redundancy, or security. Recursive name servers are used by clients to resolve DNS requests without having to perform iterative queries of root, TLD, and registrant name servers. When a recursive name server receives a query that it cannot resolve independently, it queries the upper-level zones to find the answer that it returns to the original client. This process can be sped up significantly by caching results, though care must be observed to prevent serving stale records from a cache.

DNS AND ANYCAST

- http://dyn.com/blog/unicast-vs-anycast-dns-nameserver-routing/
- http://blog.catchpoint.com/2015/06/16/dns-anycast/

PROTOCOL

As the prior sections have suggested, the DNS protocol is a query response protocol that is used to resolve IP addresses and other metadata stored in resource records based on DNS names. The protocol can be run atop both UDP and TCP for transport and uses port 53 in both cases. Basic DNS queries are most likely to be served over UDP, as this protocol provides the most efficient transport mechanism by avoiding the multi-round trip cost associated with the TCP handshake. The connection-oriented TCP protocol is most likely to be used for answers that do not fit within the constraints of a single datagram including **zone transfers** that transfer data between primary and secondary name servers.

The DNS protocol itself is unicast, though a multicast version (mDNS) has been defined to support communication on local networks without the need for supporting infrastructure. Multicast DNS is served on port 5353 of the 224.0.0.251 multicast address. The mDNS protocol was popularized by Apple under the Bonjour trademark, but it has more recently made its way into the most popular mobile and desktop operating systems by way of zeroconf initiatives.

COMMON RESOURCE RECORD (RR) TYPES

DNS-related RFCs define many different record types for general use and specific applications. We address the most common records here.

- https://blog.dnsimple.com/2015/04/common-dns-records/

Start of Authority (SOA) records identify the primary name server and administrator responsible for a given zone as well as version information and base parameters that impact communication between servers in a zone and caching of records related to the zone.

Name Server (NS) records indicate which name servers are authoritative for a given zone. These records are also accompanied by so-called glue records that store the IP addresses associated with the server name stored in an NS.

Address (A) records are among the most familiar DNS records. Their role is simple in that they associate the IPv4 address of a resource with the server's name. Address records are structured for forward lookup, i.e., resolving the address given the name. A similar, yet separate address record (AAAA) is defined to support IPv6-related queries.

Canonical Name (CNAME) records are used to set aliases for existing DNS names rather than directly associating a name with an IP address. These records are commonly used when a resource is hosted by an organization other than the registrant itself and to avoid the necessity of maintaining static IP addresses for these resources. The hostname of a web site that runs on a service like SquareSpace or Wix is likely to point to a CNAME alias that can be resolved to an IP within the parent (service-provider's) domain.

Reverse DNS Pointer (PTR) records enable lookups from IPv4 address to hostnames. These reverse lookups are required, or at least recommended, for the proper operation of certain application-level protocols such as email.

Text (TXT) records store additional information required by other protocols. Quite often, TXT records are created to demonstrate ownership of a certain namespace. This action is required, e.g., when attaching a private domain to a hosted service such as Office 365 or GSuite mail. TXT records are also used to store Sender Policy Framework (SPF) records leveraged by SMTP servers to combat email spam.

Mail Exchange (MX) records point to SMTP servers for a domain, enabling other SMTP servers to locate and route mail between domains. Each MX record contains the FQDN for a mail host. An associated A (or AAAA) record must exist to resolve the hostname to an IP address.

The MX record also contains a preference field, which is used to determine priority among multiple MX records. It is quite common to provide multiple MX records to balance load between mail servers and gain redundancy. Servers with smaller preference values are tried first, while clients are expected to choose randomly between servers with equal preference.

The MX record is the starting point for mail configuration. To combat spam and fraud, most organizations will configure SMTP to check for additional records, such as the TXT-based SPF records. Read https://www.rackaid.com/blog/email-dns-records/ to learn more about these applications.

ADDITIONAL TOPICS

TOOLS

- Dig - https://help.dyn.com/how-to-use-binds-dig-tool/
- Nslookup - https://en.wikipedia.org/wiki/Nslookup
- Whois - https://whois.icann.org/en/dns-and-whois-how-it-works

SERVICE DISCOVERY

Service (SRV) records are used in conjunction with application layer protocols to identify resources based on role or function rather than hostname. This application is known as Service Discovery (DNS-SD) and is described in RFC 6763.

SRV records can be queried just like any other record, though they are identified by a distinct convention of _service._transport in place of a standard host name. Practically speaking, SRV records function much like a general purpose MX pointing to named address records that can be further queried to obtain the IP address for a resource.

```
$ dig -t srv _sip._tls.microsoft.com

; <<>> DiG 9.8.3-P1 <<>> srv _sip._tls.microsoft.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33094
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0,
ADDITIONAL: 1

;; QUESTION SECTION:
;_sip._tls.microsoft.com.      IN    SRV

;; ANSWER SECTION:
_sip._tls.microsoft.com. 3600 IN    SRV   0 0 443
sip.microsoft.com.

;; ADDITIONAL SECTION:
sip.microsoft.com.   895   IN    A     167.220.67.163

;; Query time: 15 msec
;; SERVER: 199.66.140.50#53(199.66.140.50)
;; WHEN: Tue May  9 16:19:39 2017
;; MSG SIZE  rcvd: 98
```

The preceding query highlights the use of SRV by the session initiation protocol (SIP), a component in voice-over-IP and other types of real-time communication systems that is used to establish connections between peers.

- https://en.wikipedia.org/wiki/SRV_record
- https://www.onsip.com/blog/dns-srv-records-sip

SECURITY AND PRIVACY

The distributed nature of DNS infrastructure offers many benefits. The database is independent of any single network or entity. The load of DNS queries is distributed broadly across many different servers. Clients can query servers that are geographically close to them, reducing overall time for name resolution. Moreover, DNS can withstand many types of failures and disruptions and resists many types of broad scale interference.

Nevertheless, DNS has some significant limitations in the realm of security and privacy that are important to understand. In standard DNS, queries and responses are neither encrypted nor authenticated. Targeted manipulation of DNS responses, e.g., spoofing, can be used to direct victims to attacker-controlled resources or to inject advertisements into user traffic.

- http://erichelgeson.github.io/blog/2013/12/31/i-fought-my-isps-bad-behavior-and-won/

Likewise, passive monitoring reveals significant amounts of information about a victim's identity, beyond just telling us which server they have visited. This type of monitoring is often conducted by network administrators, ISPs, governments, passive observers on wireless networks, and hackers who have managed to hijack network traffic through a MITM attack.

- https://nakedsecurity.sophos.com/2016/10/05/unmasking-tor-users-with-dns/
- https://dnsprivacy.org/wiki/display/DP/DNS+Privacy+-+The+Problem

This is a double-edged sword. The same properties of DNS can make it a powerful tool for detecting and preventing malicious attacks. The Cisco Umbrella service (original OpenDNS), for example, can detect malware infections, phishing scams, and more through sophisticated analysis of DNS queries (https://umbrella.cisco.com/).

One approach to securing DNS against privacy is to use a resolver that encrypts connections between clients and recursive servers. The dnscrypt project (https://dnscrypt.org/) provides such a tool, though it is not supported by the vast majority of public DNS servers.

More than privacy, the integrity of DNS is arguably the primary concern for most applications. DNSSEC provides a basic integrity mechanism akin to the public key infrastructure and X.509 certificates that underly SSL. DNSSEC is a complex topic and quite underappreciated. Fortunately, many organizations can derive its benefits simply by enabling the feature within a third-party DNS host.

- https://www.cloudflare.com/dns/dnssec/how-dnssec-works/

At a more focused level, the integrity of DNS is contingent on organizations to protect records under their purview. Basic DNS security hygiene involves the following tasks:

- Choosing trustworthy and secure DNS providers
- Maintaining DNS registrations to prevent attackers from hijacking the domain
- Limiting access to zone management accounts

- Implementing strong passwords and secondary authentication mechanisms

It's difficult to overstate the importance of these practices. Attackers target DNS to hijack web traffic, entire email systems, and more. For this reason, applications are often deployed with higher-layer integrity mechanisms, e.g., X.509 certificates used in SSL and fingerprinting / key pinning in SSH. These systems, however, are not foolproof. Each has its limitations, therefore DNS security remains a top priority.