Introduction to the Domain Name System (DNS)

Originalartikel

Backup

<html> <p>Surfing the web is fun and easy, but think what it would be like if you had to type in the IP address of every website you wanted to view. For example, locating a website would look like this when you type it in: <a href=„https://54.204.39.132“>https://54.204.39.132</a>, which would be nearly impossible for most of us to remember. Of course, using bookmarks would help, but suppose your friend tells you about a cool new website and tells you to go to 54.204.39.132. How would you remember that? Telling someone to go to „Opensource.com“ is far easier to remember. And, yes, that is our IP address.</p><p>The Domain Name System provides the database to be used in the translation from human-readable hostnames, such as <a href=„http://www.opensource.com“>www.opensource.com</a>, to IP addresses, like 54.204.39.132, so that your internet-connected computers and other devices can access them. The primary function of the <a href=„https://www.isc.org/downloads/bind/“ target=„_blank“>BIND</a> (Berkeley Internet Name Domain) software is that of a domain name resolver that uses that database. There is other name resolver software, but BIND is currently the most widely used DNS software on the internet. I will use the terms name server, DNS, and resolver pretty much interchangeably throughout this article.</p> <p>Without these name resolver services, surfing the web as freely and easily as we do would be nearly impossible. As humans, we tend to do better with names like Opensource.com, while computers do much better with numbers like 54.204.39.132. Therefore, we need a translation service to convert the names that are easy for us to the numbers that are easy for our computers.</p> <pre> <small readability=„1“> <div class=„geshifilter“ readability=„8“><pre class=„bash geshifilter-bash“>127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4 ::1 localhost localhost.localdomain localhost6 localhost6.localdomain6 &#160; # Lab hosts 192.168.25.1 server 192.168.25.21 host1 192.168.25.22 host2 192.168.25.23 host3 192.168.25.24 host4</pre></div></small></pre><p>In small networks, the <strong>/etc/hosts</strong> file on each host can be used as a name resolver. Maintaining copies of this file on several hosts can become very time-consuming and errors can cause much confusion and wasted time before they are found. I did this for several years on my own network and it eventually became too much trouble to maintain even with only the usual eight to 12 computers I usually have operational. Ultimately, I converted to running my own name server to resolve both internal and external hostnames.</p> <p>Most networks of any size require centralized management of this service with name services software such as BIND. Hosts use the Domain Name System (DNS) to locate IP addresses from the names given in software such as web browsers, email clients, SSH, FTP, and many other internet services.</p> <h2>How a name search works</h2> <p>Let's take look at a simplified example of what happens when a name request for a web page is made by a client service on your computer. For this example, I will use <a href=„http://www.opensource.com/“ target=„_blank“>www.opensource.com</a> as the website I want to view in my browser. I also assume that there is a local name server on the network, as is the case with my own network.</p> <p>1. First, I type in the URL or select a bookmark containing that URL. In this case, the URL is <a href=„http://www.opensource.com“>www.opensource.com</a>.</p> <p>2. The browser client, whether it is Opera, Firefox, Chrome, Lynx, Links, or any other browser, sends the request to the operating system.</p> <p>3. The operating system first checks the <strong>/etc/hosts</strong> file to see if the URL or hostname is there. If so the IP address of that entry is returned to the browser. If not, I proceed to the next step. In this case, I assume that it is not.</p> <p>4. The URL is then sent to the first name server specified in <strong>/etc/resolv.conf</strong>. In this case, the IP address of the first name server is my own internal name server. For this example, my name server does not

have the IP address for <a href=„http://www.opensource.com“>www.opensource.com</a> cached and must look further afield. Let's go on to the next step.</p> <p>5. The local name server sends the request to a remote name server. This can be one of two destination types, one type of which is a forwarder. A forwarder is simply another name server, such as the ones at your ISP, or a public name server, such as Google at 8.8.8.8 or 8.8.4.4. The other destination type is that of the top-level <a href=„https://en.wikipedia.org/wiki/Root_name_server“ target=„_blank“>root name servers</a>. The root servers don't usually respond with the desired target IP address or <a href=„http://www.opensource.com“>www.opensource.com</a>, they respond with the authoritative name server for that domain. The authoritative name servers are the only ones that have the authority to maintain and modify name data for a domain.</p> <p>The local name server is configured to use the root name servers so the root name server for the .com top-level domain returns the IP Address of the <a href=„https://en.wikipedia.org/wiki/Name_server#Authoritative_name_server“ target=„_blank“>authoritative name server</a> for <a href=„http://www.opensource.com“>www.opensource.com</a>. That IP address could be for any one of the three (at the time of this writing) name servers, ns1.redhat.com, ns2.redhat.com, or ns3.redhat.com.</p> <p>6. The local name server then sends the query to the authoritative name server, which returns the IP address for <a href=„http://www.opensource.com“>www.opensource.com</a>.</p> <p>7. The browser uses the IP address for <a href=„http://www.opensource.com“>www.opensource.com</a> to send a request for a web page, which is downloaded to my browser.</p> <p>One of the important side effects of this name search is that the results are cached for a period of time by my local name server. That means that the next time I, or anyone on my network, wants to access Opensource.com, the IP Address is probably already stored locally, which prevents doing a remote lookup.</p> <h2>The DNS database</h2> <p>The DNS system is dependent upon its database to perform lookups on hostnames to locate the correct IP address. The DNS database is a general-purpose distributed, hierarchical, replicated database. It also defines the style of hostname used on the internet, properly called a FQDN (Fully Qualified Domain Name).</p> <p>FQDNs consist of complete hostnames such as hornet.example.com and test1.example.com. FQDNs break down into three parts.</p> <p>1. The TLDN (<a href=„https://en.wikipedia.org/wiki/Top-level_domain“ target=„_blank“>Top-Level Domain Names</a>), such as .com, .net, .biz, .org, .info, .edu, and so on, provide the last segment of a FQDN. All TLDNs are managed on the root name servers. Aside from country top-level domains such as .us, .uk, and so on, there were originally only a few main top-level domains. As of February 2017, there are 1528 top-level domains.</p> <p>2. The second level domain name is always immediately to the left of the top-level domain when specifying a hostname or URL, so names like Redhat.com, Opensource.com, Getfedora.org, and example.com provide the organizational address portion of the FQDN.</p> <p>3. The third level of the FQDN is the hostname portion of the name, so the FQDN of a specific host in a network would be something like host1.example.com.</p> <p class=„rtecenter“><img alt=„wg61vm.png“ src=„https://opensource.com/sites/default/files/images/business-uploads/wg61vm.png“ width=„800“ height=„421“/></p> <p>Figure 1 shows a simplified diagram of the DNS database hierarchy. The „top“ level, which is represented by a single dot (.) has no real physical existence. It is a device for use in DNS zone file configuration to enable an explicit end stop for domain names. A bit more on this later.</p> <p>The true top level consists of the root name servers. These are a limited number of servers that maintain the top-level DNS databases. The root level may contain the IP addresses for some domains, and the root servers will directly provide those IP addresses where they are available. In other cases, the root servers provide the IP addresses of the authoritative server for the desired domain.</p> <p>For example, assume I want to browse <a href=„http://www.opensource.com“>www.opensource.com</a>. My browser makes the request of the local name server, which does not contain that IP address. My local name server is configured to use the root servers when an address is not found in the local cache, so it sends the request for <a href=„http://www.opensource.com“>www.opensource.com</a> to one of the root servers. Of course, the local name server must know how to locate the root name servers so it uses the <strong>/var/named/named.ca</strong> file, which contains the names and IP addresses of the root name servers. The <strong>named.ca</strong> file is also known as the hints file.</p> <p>In this example, the IP address for <a href=„http://www.opensource.com“>www.opensource.com</a> is not stored by the root servers. The root server uses its database to locate the name and IP address of the authoritative name server for <a href=„http://www.opensource.com“>www.opensource.com</a>.</p> <p>The local name server queries the authoritative name server, which returns the IP address of <a href=„http://www.opensource.com“>www.opensource.com</a>. The local name server then responds to the browser's request and provides it with the IP address. The authoritative name server for Opensource.com contains the zone files for that domain.</p> <pre> <small readability=„6“><div class=„geshifilter“ readability=„18“><pre class=„bash geshifilter-bash“># dig www.opensource.com &#160; ; &lt;&lt;&gt;&gt; DiG 9.10.4-P6-RedHat-9.10.4-4.P6.fc25 &lt;&lt;&gt;&gt; www.opensource.com ;; global options: +cmd ;; Got answer: ;; -&gt;&gt;HEADER&lt;&lt;- opcode: QUERY, status: NOERROR, id: 54308 ;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 4 &#160; &#160; ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;www.opensource.com. IN A &#160; &#160; ;; ANSWER SECTION: www.opensource.com. 300 IN CNAME opensource.com. opensource.com. 300 IN A 54.204.39.132 &#160; &#160; ;; AUTHORITY SECTION: opensource.com. 129903 IN NS ns1.redhat.com. opensource.com. 129903 IN NS ns3.redhat.com. opensource.com. 129903 IN NS ns2.redhat.com. &#160; &#160; ;; ADDITIONAL SECTION: ns2.redhat.com. 125948 IN A 209.132.183.2 ns3.redhat.com. 125948 IN A 66.187.233.212 ns1.redhat.com. 125948 IN A 209.132.186.218 &#160; &#160; ;; Query time: 71 msec ;; SERVER: 192.168.0.51#53(192.168.0.51) ;; WHEN: Sat Mar 04 21:23:51 EST 2017 ;; MSG SIZE rcvd: 186</pre></div></small></pre><p>Listing 2, above, shows the results of a <strong>dig</strong> command displaying not only the IP address of the desired host, but it also shows the authoritative servers, their IP addresses, and the server that actually fulfilled the request. Here is the use of the <strong>dig</strong> command that obtains the DNS information for <a href=„http://www.opensource.com“>www.opensource.com</a>. The <strong>dig</strong> command is a powerful tool that can tell us a lot of information about the DNS configuration for a host. The dig command returns the actual records from the DNS database and displays the results in four main sections. Refer to Listing 2 as you read the descriptions of these sections.</p> <p>The first section is the question section. For this example, it states that I am looking for the A record of „<a href=„http://www.opensource.com“>www.opensource.com</a>.“ Notice the dot at the end of the top-level domain name. This indicates that .com is the final domain name component in the hostname.</p> <p>The answer section shows two entries, a CNAME record and an A record. A records are the primary name resolver records and there must be an A record, which contains the IP address for each host. CNAME stands for Canonical Name and this record type is an alias for the A record and points to it. It is not typical practice to use „www“ as the hostname for a web server. It is common to see a CNAME record that points to the A record of the FQDN; however, that is not quite the case here. Notice that the A record for Opensource.com does not have a hostname associated with it. It is possible to have a record that applies to the domain as a whole as is the case here.</p> <p>The authority section lists the authoritative name servers for the Opensource.com domain. In this case, those are the Red Hat name servers. Notice that the record type is NS for these entries.</p> <p>The additional section lists the A records for the Red Hat name servers.</p> <p>Following the additional section, I can find some additional interesting information, including the IP address of the server that returned the information shown in the results. In this case, it was my own internal name server.</p> <h2>DNS Client configuration</h2> <p>Most computers need little configuration to enable them to access name services. It usually consists of adding the IP addresses of one to three name servers to the <strong>/etc/resolv.conf</strong> file. This is typically performed at boot time on most home and laptop computers because they are configured using the DHCP (Dynamic Host Configuration Protocol), which provides them with their IP address, gateway address, and the IP addresses of the name servers.</p> <p>For hosts that are configured statically, the <strong>resolv.conf</strong> file is usually generated during installation from information entered by the sysadmin doing the install. In current Red Hat-based distributions and others that use NetworkManager to manage network configuration and to perform connection management, the static information, such as name servers, gateway, and IP address, are all stored in the interface configuration files located in <strong>/etc/sysconfig/network-scripts</strong>.</p> <p>Overriding the defaults provided to a host by adding that information to the interface configuration file is possible. For example, I sometimes add my preferred name servers to the interface configuration files on my laptop and netbook. Many of the name servers provided by remote connections in public places, such as hotels, coffee shops, and even friends' personal Wi-Fi connections, can be unreliable and in some cases can use forwarders that intentionally censor results or redirect queries to pages of advertisements, so I always insert the Google public name servers in my interface configuration files. Refer to my article <em><a href=„https://opensource.com/life/16/6/how-configure-networking-linux“ target=„_blank“>How to configure networking in Linux</a></em> for information about the interface configuration files.</p> <p>Also, be aware that NetworkManager creates an interface configuration file for each Wi-Fi network it connects with. The configuration files are named for the SSID (Service Set IDentifier) of the network. Be sure to add the desired name server entries to the correct file.</p> <p>Some of the interface configuration files that have been created on my laptop by NetworkManager in the last few months are listed below.</p> <ul><li>ifcfg-enp0s25 (This is the configuration file for the wired network.)</li> <li>ifcfg-FBI-DHS.TF1_EXT</li> <li>ifcfg-HOME-14A2</li> <li>ifcfg-linksys</li> <li>ifcfg-LinuxDude</li> <li>ifcfg-MomsPlace</li> <li>ifcfg-FBI-van</li> <li>ifcfg-PointSourceGuest</li> <li>ifcfg-Red_Hat_Guest</li> <li>ifcfg-Sands_3_hoa1</li> <li>ifcfg-Sheraton_Raleigh_Guest_Access</li> <li>ifcfg-SM-CLC1</li> <li>ifcfg-xfinityWi-Fi</li> </ul><p>And no, I do not have any connection with the FBI. Someone I know who shall remain nameless has an interesting sense of humor and enjoys making the neighbors nervous.</p> <h2>DNS record types</h2> <p>There are a number of different DNS record types, and I want to introduce some of the more common ones here. My next article will describe how to create your own name server using BIND and will use many of these record types to build your name server. These record types are used in the zone files that comprise the DNS database.</p> <p>One common field in all of these records is „IN,“ which specifies that these are INternet records.</p> <p>View a complete <a href=„https://en.wikipedia.org/wiki/List_of_DNS_record_types“ target=„_blank“>list of DNS record types on Wikipedia</a>.</p> <h3>SOA</h3> <p>SOA is the Start of Authority record. It is the first record in any forward or reverse zone file, and it identifies this as the authoritative source for the domain it describes. It also specifies certain functional parameters. A typical SOA record looks like the sample below.</p> <pre> <small readability=„1“> <div class=„geshifilter“ readability=„8“><pre class=„bash geshifilter-bash“>@ IN SOA epc.example.com root.epc.example.com. (&#160;

```
                        2017031301      ; serial
                        1D              ; refresh
                        1H              ; retry
                        1W              ; expire
                        3H )            ; minimum</pre></div></small></pre><p>The
first line of the SOA record contains the name of the server for the zone
and the zone administrator, in this case root.</p>
```

<p>The second line is a serial number. In this example, I use the date in YYYYMMDDXX format where XX is a counter. Thus, the serial number in the SOA record above represents the first version of this file on March 13, 2017. This format ensures that all changes to the serial number are incremented in a numerically sequential manner. Doing this is important because secondary name servers, also known as slave servers, only replicate from the primary server when the serial number of the zone file on the primary is greater than the serial number on the secondary. Be sure to increment the serial number when you make changes or the secondary server will not sync up with the modified data.</p> <p>The rest of the SOA record consists of various times that secondary servers should perform a refresh from the primary and wait for retries if the first refresh fails. It also defines the amount of time before the zone's authoritative status expires.</p> <p>Times all used to be specified in seconds, but recent versions of BIND allow other options defined with W=week, D=day, H=hour, and M=minute. Seconds are assumed if no other specifier is used.</p> <h3>$ORIGIN</h3> <p>The $ORIGIN record is like a variable assignment. The value of this variable is appended by the BIND program to any name in an A or PTR record that does not end in a period (.) in order to create the FQDN (Fully Qualified Domain Name) for that host. This makes for less typing because the zone administrator only has to type the host name portion and not the FQDN for each record.</p> <pre> <small>

geshifilter-bash"

```
$ORIGIN &#160; &#160; &#160; &#160; example.com.
```

</small></pre><p>Also, the @ symbol is used as a shortcut for this variable and any occurrence of @ in the file is replaced by the value of $ORIGIN.</p> <h3>NS</h3> <p>The NS record specifies the authoritative name server for the zone. Note that both names in this record end with periods so that „.example.com“ does not get appended to them. This record will usually point to the local host by its FQDN.</p> <pre> <small>

geshifilter-bash"

```
example.com. &#160; &#160; &#160; &#160; &#160; &#160;IN &#160; &#160; &#160;NS &#160; &#160; &#160;epc.example.com.
```

</small></pre><p>Note that the host, epc.example.com, must also have an A record in the zone. The A record can point to the external IP address of the host or to the localhost address, 127.0.0.1.</p> <h3>A</h3> <p>The A record is the Address record type that specifies the relationship between the host name and the IP address assigned to that host. In the example below, the host test1 has IP address 192.168.25.21. Note that the value of $ORIGIN is appended to the name test1 because test1 is not an FQDN and does not have a terminating period in this record.</p> <pre> <small>

geshifilter-bash"

```
test1 &#160; &#160; &#160; &#160; &#160; &#160; &#160; &#160; &#160; IN &#160; &#160; &#160;A &#160; &#160; &#160; 192.168.25.21
```

</small></pre><p>The A record is the most common type of DNS database record.</p> <h3>CNAME</h3> <p>The CNAME record is an alias for the name in the A record for a host. For

example, the hostname server.example.com might serve as both the web server and the mail server. There would be one A record for the server, and possibly two CNAME records as shown below.</p> <pre> <small readability=„0“><div class=„geshifilter“ readability=„6“><pre class=„bash geshifilter-bash“>server IN A 192.168.25.1 www IN CNAME server mail IN CNAME server</pre></div></small></pre><p>Lookups with the <strong>dig</strong> command on <a href=„http://www.example.com“>www.example.com</a> and mail.example.com will return the CNAME record for mail or www and the A record for server.example.com.</p> <h3>PTR</h3> <p>The PTR records are to provide for reverse lookups. This is when you already know the IP address and need to know the fully qualified host name. For example, many mail servers do a reverse lookup on the alleged IP address of a sending mail server to verify that the name and IP address given in the email headers match. PTR records are used in reverse zone files. Reverse lookups can also be used when attempting to determine the source of suspect network packets.</p> <p>Be aware that not all hosts have PTR records, and many ISPs create and manage the PTR records, so reverse lookups may not provide the needed information.</p> <h3>MX</h3> <p>The MX record defines the Mail eXchanger, (i.e., the mail server for the domain example.com). Notice that it points to the CNAME record for the server in the example above. Note that both example.com names in the MX record terminate with a dot so that example.com is not appended to the names.</p> <pre> <small readability=„0“><div class=„geshifilter“ readability=„6“><pre class=„bash geshifilter-bash“>; Mail server MX record example.com. IN MX 10 mail.example.com.</pre></div></small></pre><p>Domains may have multiple mail servers defined. The number „10“ in the above MX record is a priority value. Other servers may have the same priority or different ones. The lower numbers define higher priorities. Therefore, if all mail servers have the same priority they would be used in round-robin fashion. If they have different priorities, the mail delivery would first be attempted to the mail server with the highest priority&#8212;the lowest number&#8212;and if that mail server did not respond, delivery would be attempted to the mail server with the next highest priority.</p> <h3>Other records</h3> <p>There are other types of records that you may encounter in the DNS database. One type, the TXT records, are used to record comments about the zone or hosts in the DNS database. TXT records can also be used for DNS Security. The rest of the DNS record types are outside the scope of this article.</p> <h2>Final thoughts</h2> <p>Name services are a very important part of making the internet easily accessible. It binds the myriad disparate hosts connected to the internet into a cohesive unit that makes it possible to communicate with the far reaches of the planet with ease. It has a complex distributed database structure that is perhaps even unknowable in its totality, yet that can be rapidly searched by any connected device to locate the IP address of any other device that has an entry in that database.</p> <p>If you are a system administrator on a network of almost any size, you may find it helpful to know how to build your own name server. I will describe how you can do that in my&#160;next article, <a href=„https://opensource.com/article/17/4/build-your-own-name-server“ target=„_blank“>Build your own DNS server on Linux</a>.</p> <h2>Resources</h2> <ul><li><a href=„https://www.iana.org/“>IANA (Internet Assigned Numbers Authority)</a></li> <li><a href=„http://www.zytrax.com/books/dns/ch8/soa.html“>SOA (Start of Authority) record</a></li> <li><a href=„https://en.wikipedia.org/wiki/List_of_DNS_record_types“>List of DNS Record Types</a></li> <li><a href=„https://blog.dnsimple.com/2015/04/common-dns-records/“>Common DNS records and their uses</a></li> </ul> </html>