Oversimplified DNS

... or, even a rocket scientist can understand DNS

Step 1 - Verify WHOIS information

GOALS: Make sure that WHOIS reports every name server you have, and doesn't report any that aren't authoritative for your domain. Also, start a list of your nameservers to test in Step 2.

BACKGROUND: When there may be a problem with your DNS, the first step is to check out your WHOIS listing. This is the first step in the DNS process. You get a domain name (say, www.fcc.com) by going to the responsible party for the top level domain you want to be in (.com). For the most popular top level domains (such as .com), you go through the Internic, or someone they have authorized.

The Internic (or other appropriate auhority) keeps a "WHOIS" database, which includes your contact information, as well as 2 (or more) DNS servers that will handle your domain. The DNS servers listed in WHOIS are the start of the DNS process. Those get sent to the root DNS servers, and make sure that anyone on the Internet can do DNS lookups on your domain. When someone looks up one of your domains, the first step (assuming that there is not a cached entry) is to contact one of the root servers to find a name server for your domain.

Step 1a: Find your listing

There are a number of ways to find your WHOIS listing. First, you can try a WHOIS program that runs on your computer (Sam Spade is a great Windows freeware program that can do this). Or, you can go to a web site such as http://www.whois.net/, or http://www.samspade.org/, or rs.internic.net.

Step 1b: Check the contact information for accuracy

If there is an error in the contact information (such as your E-mail address), it is possible that if there are problems with your domain, it will take longer for you to find out.

Step 1c: Write down all the nameservers listed in WHOIS.

Later, we will check them to make sure that all of them are working and authoritative for your domain, and that you don't have extra nameservers not listed here. *If there is only one server listed here, you have a serious problem -- contact the company you got your domain name from and get another name server listed in WHOIS.*

Step 2 - Find All Name Servers

GOALS: Find every name server that may be authoritative for your domain. Make sure that all are authoritative, and that WHOIS has a list of all of them (and no others).

BACKGROUND: Every domain name is required to have a primary nameserver (only one), and at least one secondary nameserver. This is to help ensure that all domains are reachable. For example, if you have 3 separate offices, and one of them suffers a disaster, that shouldn't prevent people from being able to reach the other two offices.

The WHOIS database lists all the authoritative nameservers (ones that get updated automatically when you make changes) for your domain. This list gets sent to the root server for your Top Level Domain (such as .com). Anyone in the world trying to reach your domain will get that list, and go to one of the nameservers to resolve a subdomain within your domain.

Step 2a: Find out what the root servers think your nameservers are

To do this, you will need to use a program that can generate DNS requests (such as NSLOOKUP on Windows or DIG). My favorite program for this is Sam Spade. Or, you can go to a web site that allows these requests, such as http://www.samspade.org/.

First, find the primary root server for your Top Level Domain (such as .com if you have the domain example.com). With NSLOOKUP, you can type "set type=NS" (to get the NS, or nameserver, records) and then enter your top level domain ("com" or "uk" or whatever). You will get a list of root nameservers. Type "server " followed by the first nameserver in the list (for example, "server a.root-servers.net"). Then, enter your domain name ("example.com"). Now, you will have a list of servers that the root servers think are authoritative for your domain.

With DIG, enter your top level domain ("com") as the domain to look up (or type "dig com@default.dnsserver" where "default.dnsserver" is your normal DNS server). Look at the "NS" entries in the results, and take the top one, and enter it as the nameserver to use. Then, as the domain to look up, enter your domain name (or, type "dig example.com@a.root-servers.net", where your domain goes before the "@" and the root server goes after the "@"). This should just return the name servers for your domain, but make sure just to look at entries with " NS " in them.

Problem? If there are any nameservers listed here that are NOT listed in WHOIS, or there are any nameservers listed in WHOIS that are NOT listed here, there is a problem. Contact the company that gave you your domain name for help.

Add these to your list of nameservers from WHOIS that you got in Step 1.

Step 2b: Find out what your name servers say your nameservers are

For this step, do exactly what you did in Step 2a, except this time use the first one of the name servers you have written down as the DNS server to use, and use your domain name as the domain to look up.

Using NSLOOKUP, type "server nameserver.example.com" (where nameserver.example.com is the first name server on the list you wrote down), then "set type=ns", and then enter your domain name. If there are any name servers listed here that are not on your list, add them to the list.

Or, using DIG, enter the first name server you wrote down as the name server to use, and then your domain name as the domain to look up. Or, from the command line, type "dig example.com@nameserver.example.com". If there are any name servers listed here that are not on your list, add them to the list (only look at entries with " NS " in them).

Problem? If any name server you check here returns a name server that was NOT listed in WHOIS, it is called a "missing nameserver", and is an error. If the name server listed is authoritative for your domain (see steps below), it MUST (RFC882 p.25) be added to the WHOIS listing. If it is NOT authoritative for your domain, it MUST be taken out of the nameserver it was listed in.

Problem? If any name server you check here does NOT return one of the name servers listed in WHOIS, there is a problem. If the name server listed in WHOIS IS authoritative for your domain (see steps below), it MUST be added to your nameservers. If it is NOT authoritative, it MUST be taken out of WHOIS (but you MUST have at least 2 nameservers listed in there). *Repeat this step for **EVERY** name server on your list -- even ones you may have just added.*

Step 2c: Test every name server

You need to make sure that every name server on your list is authoritative for your domain.

With DIG, use the first name server on your list as the domain server to use, and enter your domain name as the domain to look up (or, from the command line, type "dig example.com@nameserver.example.com"). You should see "Authoritative answer" (or something similar) if it is authoritative, or "Non-authoritative answer" if it is not.

Problem? If the name server is NOT authoritative for your domain, you have a "lame delegation." If this server is not authoritative, and it is listed as an NS record in your domain (step 2b), it MUST be taken out. If this server is not authoritative, and it is listed in WHOIS, it MUST be taken out (but, you MUST also have at least 2 name servers listed in WHOIS).

Step 2d: Sanity Checks

- You MUST have ONLY ONE primary nameserver. You will check this later in the step for SOA (Start of Authority) records, where the primary server is listed.
- You MUST have AT LEAST ONE secondary nameserver. That means that WHOIS, the root servers, and all your name servers MUST list at least two name servers that are authoritative for your domain. You may have more than one secondary nameserver (4 to 7 SHOULD be the maximum).
- Your secondary nameserver(s) MUST get their information directly from your primary nameserver, and check it periodically (how often is based on the SOA record fields) for updates.

Step 3 - Verify SOA (Start of Authority)

GOAL: Find your SOA record and make sure that it is accurate.

BACKGROUND: The SOA record has core information about your zone. It defines which server is your primary nameserver, your contact information (E-mail), how your secondary nameservers get updated, and the default (minimum) Time-To-Live values for your records.

Step 3a: Get your SOA data

To get your SOA data, you can use NSLOOKUP or DIG (or any other program, or a web site, that can query DNS records from a nameserver you choose). You need to use your domain as the domain name to query, query for the SOA record, once for each nameserver on the list you wrote down.

Using NSLOOKUP, enter "server nameserver.example.com" (do this once for each nameserver on your list, replacing "nameserver.example.com" with one nameserver at a time). Then, type "set type=SOA". Finally, type your domain name ("example.com"). You will see the SOA record for your domain.

Using DIG, enter your domain name as the domain name to query, and enter each server from the list you wrote down (one at a time), and look at the SOA or "Zone of Authority" section.

Problem? Make sure that the SOA record returned by each name server is *identical*. If the serial numbers are different, you will have to wait up to the number of seconds listed in the "refresh" section of the nameserver with the lower serial number for it to get updated (or more time if the secondary nameserver can't reach the primary). If the primary nameserver has a lower serial number than a secondary, you have a serious problem that you will need to fix. If the serial numbers are the same, but other data is different, you have a serious problem -- your primary was updated without updating the serial number (update the serial number and the problem will get fixed).

Problem? The SOA record MUST be the FIRST record in your zone file, and MUST also be the LAST record in your zone file. It must appear only those two times, and both of the entries must be identical. This can be verified correctly only on the nameserver itself; the procedure varies depending on the software you use.

Step 3b: Check your SOA data

- **MNAME** ("Primary NS") - This entry is the domain name of the name server that was the original source of the data (this entry MUST be your primary nameserver). This is your primary nameserver, and MUST be the one and only server that you ever update. You must not update the secondary server(s) -- they will update automatically, based on this the SOA record. **Problem?** This should be a fully qualified domain name .
- **RNAME** ("Responsible Person") - This is a **DOMAIN NAME** that indicates the E-mail address of the person responsible for this zone. It MUST be in the format username.domain.tld; IE "jimmy.example.com" if the E-mail address is "jimmy@example.com". **Problem?** If this record has an "@" sign in it, it is wrong! (some programs, such as Sam Spade, may put the "@" sign in there when displaying it -- check the actual SOA record to be sure). **Problem?** If this record doesn't have a domain name in it (ie, just "hostmaster"), *it is wrong!*. It is recommended [RFC1912 2.2] to use the format "hostmaster.example.com", of course making sure that "hostmaster@example.com" is a valid E-mail address! If the E-mail address has a "." in it, there must be a "\" before it (for example, "scott\.perry.example.com" for "scott.perry@example.com"). **Problem?** Make sure that there is no "@" sign in this entry, otherwise there is an error.
- **SERIAL** - This is a serial number (32-bit unsigned integer) that must be *incremented* on the primary name server whenever a change is made. The recommended [RFC1912 2.2] value for this is a 10-digit number in the form YYYYMMDDnn (year, month, date, revision). For example, if you change the primary on June 4, 2001, you would enter 2001060401. If you change it again that day, you would enter 2001060402. Using this format (rather than 1, 2, 3, 4, ...) is very useful, as you can determine the last day the file was changed (which comes in handy when looking at cached entries in other DNS servers). It can also be used to double-check that you remembered to update the serial number when you last made a change. **Problem?** Make sure that there is NOT a decimal point in the serial number -- if so, it won't work as you think [RFC1912 2.2].
- **REFRESH** - The number of seconds (32-bit integer) between the time that a secondary name server gets a copy of the zone (or sees that it hasn't changed), and the next time it checks to see if it needs a new copy. This should be set to the amount of time you think it is O.K. for your secondary to have out-of-date information when you update your primary server. An hour or two might be a good value. If set too short (say, 1 minute), it will cause more traffic. If set too long (say, 1 day), the secondary servers might give out old information for up to a day. **Problem?** Make sure this value isn't very high, say a week or more, or else it will take a long time for your secondary nameservers to update when you make changes.
- **RETRY** - The number of seconds (32-bit integer) that the primary name server(s) should wait, if an attempt to refresh failed, before making another attempt to refresh. If your primary nameserver is reliable, this value should never be needed.
- **EXPIRE** - The number of seconds (32-bit integer) that lets the secondary name server(s) know how long they can hold the information before it is no

longer considered authoritative. A good value might be 2 to 4 weeks [RFC1912 2.2]. It should be long enough to keep the data during a major outage. **Problem?** Make sure that the value is greater than the minimum and retry intervals, or else the data will immediately expire if the secondary server can't reach the primary server.

- **MINIMUM** ("Minimum TTL") - The number of seconds that the records in the zone are valid for (time-to-live, or TTL), unless the records have a higher TTL value. This is VERY important! I would recommend setting this to one day, or less if you change your DNS often (note that RFC1912 2.2 suggests at least 3 days if your DNS is fairly stable). If you rarely change your DNS, and have a lot of traffic, you could increase this somewhat to minimize traffic. **Problem?** Make sure that this value isn't too long (say a week or more); otherwise, when you have to make a change to one of your servers, it will take this long before everyone knows about it -- and you may not have that much time. If you are about to make a DNS change, you can lower the value temporarily, wait the old default TTL, make your changes, and then raise it back again.

Step 4 - Finding and testing your A (address) records

GOAL: Make sure that your A records are working properly.

BACKGROUND: "A" Records are what DNS really boils down to. An A record gives you the IP address of a domain. That way, users that try to go to www.example.com will get to the right IP address.

Step 4a: Get a list of all your A records

There are several ways you can go about this. The best way is with a Zone Transfer. This will show you every record in your DNS. However, your nameservers may refuse zone transfers to any computer except the secondaries nameservers. In that case, you'll have to go to one of the nameservers, and get the master zone file.

To do a zone transfer in NSLOOKUP, first type "server nameserver.example.com", where "nameserver.example.com" is one of your nameservers (try the primary nameserver first, if it doesn't work, try the secondary(s)). Then, type "ls -d example.com". If you have a lot of entries, you might want to save it to disk using "ls -d example.com > filename.txt", where filename.txt is the name of the file to save to (you might not be able to choose a directory, it may place it automatically in the directory you are in). This will list all records in your zone; just pay attention to the A records for now.

To do a zone transfer in Sam Spade, go to the Tools menu, and choose "Zone Transfer" (if it is greyed out, go to the Edit menu, choose Options, then the Advanced table, and click 'Enable zone transfers'). Enter your domain as the domain to transfer zone information from. Enter your primary nameserver as the nameserver to use. If your primary nameserver refuses to do a zone transfer, try the secondary nameserver(s). Just pay attention to the A records for now.

If you can't get a zone transfer, the next step is to get the zone file from your primary nameserver (you'll may need to look at the instructions for your DNS server software to see where the file is located). Then, use NSLOOKUP, DIG, or other software to check all the A records (using your primary name server as the name server to use) for the domain names listed in the zone file to make sure that they match.

If you don't have access to your nameservers (if an ISP handles your DNS, for example), and you can't do a zone transfer or get the zone file, you should at least know the subdomains on your domain (for example, "www.example.com" and "mail.example.com"). Use NSLOOKUP, DIG, or other software to look up the A record for each of those domains (using your primary nameservers as the name server to use).

Note: If you know a subdomain exists (such as www.example.com), but there is no "A" record, that may be O.K. -- there could be a CNAME record pointing to another record that has an A record. For example, "www.example.com CNAME sparky.example.com" with a corresponding "sparky.example.com A 10.11.12.13". However, it is best to avoid using CNAMEs unless you are quite familiar with them!

Note: If you did a zone transfer, it will be helpful to save the results for Steps 5 and 6, where you check the MX and CNAME records.

Step 4b: Test all your A records

Go through each A record you found in step 4b.

First, make sure that every A record points to one and only one IP address (and not to a domain name).

Note: It is O.K. for a subdomain to have 2 or more A records (this also applies to most other types of records, as well). For example, you can have "www.example.com A 10.11.12.13" and "www.example.com A 10.11.12.14". That means that both IP addresses (10.11.12.13 and 10.11.12.14) can handle anything for www.example.com (technically, can handle anything an A record normally handles; for example, mail uses an MX record). You can not have 2 or more IP addresses on the same line (in the same record), however; they must be split into separate records.

Next, check to make sure that IP addresses are correct. Assuming that you know what each machine is used for, you should know how to test them (making sure to connect to the IP address). For example, if a machine is a web server, you can use a browser to connect (to http://10.11.12.13 for example). If the machine is an FTP server, FTP to 10.11.12.13. If you don't know what a machine is used for, you should find out! If you still don't know, at least try to use PING to make sure that the machine is responding ("ping 10.11.12.13").

Step 5 - Finding and testing your MX (mail) records

GOAL: Make sure that your MX records are working properly.

BACKGROUND: MX (Mail Exchange) records are used to have mail delivered to users on your domain. It MUST have an MX record (not just an A record), primarily because people typically use an E-mail address with your domain name ("joe@example.com"), not a subdomain ("joe@mail.example.com").

When you send mail to someone, your mail typically goes from your E-mail client to an SMTP server. The SMTP server then checks for the MX record of the domain in the E-mail address. For example, with "joe@example.com", it would look for the MX record for example.com. If a user did have an E-mail address "joe@mail.example.com", the SMTP server would look for the MX record of "mail.example.com". The MX record is a domain name, so the SMTP server then gets the A record for that domain name, and connects to the mail server.

Each MX record has 2 pieces of information associated with it. The first is a number ("Preference" number), the second is the domain name of the mail server. If there are multiple MX records, the SMTP server will pick one based on the preference level (starting with the lowest preference number, working its way up). It's O.K. to have more than one MX record with the same preference.

An example would be "example.com MX 10 mail.example.com", "example.com MX 50 mail1.myisp.com", and "example.com MX 50 mail2.myisp.com". An SMTP server would first try mail.example.com, and if that wasn't reachable, it would try either mail1.myisp.com or mail2.myisp.com (normally, it should pick one of the two randomly, unless it has a better reason to pick one over the other, since the preferences are the same).

Step 5a: Find your MX record(s)

Most domains only have one set of MX records, the MX records for their domain (and not any subdomains). Your first step is to get those MX records. If you saved the information you got in Step 4a, you should already have a list of your MX records. If not, follow the instructions in step 4a ("Get a list of all your A records"), but look for "MX" records instead of "A" records.

Step 5b: Check your MX records

[If you have MX records for more than one domain, such as "example.com" and "ihostforthem.example.com", you will need to repeat steps 5b and 5c once for every domain with MX records]

First, make sure that all the MX records for your domain point to a domain name (IE, "example.com MX 10 mail.example.com"). Next, make sure that all the mail server names from the MX records have a corresponding A record. You can check the A record as described in Step 4a. For example, if you have "example.com MX 10 mail.example.com", you must also have a record such as "mail.example.com A 10.11.12.15" (unless the mail server isn't in your domain, such as "example.com MX 10 mail.myisp.com" -- but if so, myisp.com must have an A record for mail.myisp.com).

Problem? Your MX records MUST NOT point to a CNAME record. For example, "example.com MX 10 mail.example.com" can not have a corresponding record "mail.example.com CNAME smtp.example.com".

Problem? Your MX records MUST NOT point to an IP address. If so, mail servers probably will not deliver mail to you!

Problem? Do NOT use wildcards (a "*" in a domain name) unless you are positive you know what you are doing. In most cases, they provide unexpected results [RFC1912 2.7].

Step 5c: Make sure your mail servers accept your mail

Next, connect to every mail server listed here to make sure that they exist, are responding to SMTP requests, and accept mail addressed to your domain. Every mail server you have listed MUST either be one your control, or one that has given you permission to use them [RFC 1912 2.5]

To do this, use Telnet. For the "Host Name", enter the mail server name (for example, "mail.example.com"). For the Port, enter 25. A second or two later you should see a welcome message. Type "HELO" followed by the domain name of the computer you are using. For example, "HELO eagle.example.com". Then, after you get a response, type "MAIL FROM: my.email.address@example.com" (using your E-mail address on your mail server), and then "RCPT TO: my.email.address@example.com". Then, type "DATA", "Subject: Test", a blank line, "Test", and then ".". After the response, type QUIT. Make sure that you get a copy of this E-mail. If you do not (it could take some time if you are using a slow mailserver outside your domain as a backup), try it again (in case you made a mistake). If you don't get that second test message, you probably have a serious problem (you may get an E-mail back saying the message wasn't deliverable).

Problem? If you only have 1 MX record for your domain, you really ought to add a backup mail server. You don't have to, but it will make you look more professional, especially if your mail server isn't reachable for some reason.

Step 6 - Check CNAME records

GOALS: Make sure that any CNAME records are accurate and problem free. Make sure that there are no unnecessary CNAME records that could cause problems later.

BACKGROUND: CNAME records are "canonical name" records. DNS allows machines to have a true (canonical name), as well as an unlimited number of aliases. The CNAME record takes care of aliases. *These should only be used when absolutely necessary*, unless you are *very* familiar with DNS, since they can cause lots of problems if not used properly.

One of the times where CNAME records can be useful is when you want a subdomain to point to a computer outside of your domain. For example, you might want "news.example.com" to go to your ISP's newsserver. Instead of putting in the IP

address, you could put in "news.example.com CNAME news.myisp.com", so that if the IP address of the newsserver changed, you wouldn't have to make any changes.

It is also said that CNAMEs may be useful when you are renaming a host, and will later get rid of the current name [RFC1912 2.4].

Finally, [RFC1912 2.4] suggests that CNAMES are good for generic names, for example, having "www.example.com CNAME funky.example.com", so the machine can have its own official name, but users can still find it without knowing its real name. Be careful with this though! In this case, you can have an A record for www.example.com pointing to the IP address that funky.example.com has (however, a reverse DNS lookup for the IP address can only return one of the names).

Step 6a: Find your CNAME entries

Get the CNAME information from step 4a, or repeat step 4a looking for CNAME records instead of A records. You should now have a list of all the CNAME entries for your domain.

Step 6b: Test your CNAME entries

Go through each CNAME entry, and make sure that the CNAME entry resolves correctly to an IP address. For example, if you have "news.example.com CNAME news.myisp.com", make sure that "news.myisp.com" has an A record pointing to a valid IP address. Also, make sure that the IP address responds as expected (in this case, run a news program to connect to the newsserver). See Step 4b for more information on making sure that the computer does what it is expected to, and is connected to the Internet.

Problem? Make sure that you have no unnecessary CNAME entries; they can make things confusing, and are only recommended if you have a legitimate need to have them and are quite familiar with DNS.

Problem? If you have a CNAME entry, make sure that it is the ONLY resource record for that domain. For example, if you have "www.example.com CNAME sparky.example.com", you must not have any A records, MX records, etc. for "www.example.com". [RFC1034 3.6.2] [RFC1912 2.4]

Problem? If MUST NOT have an NS record pointing to a CNAME. For example, "example.com NS dns.example.com" and "dns.example.com CNAME ns0.example.com" will cause problems [RFC1912 2.4]

Problem? If MUST NOT have an MX record pointing to a CNAME. For example, "example.com MX mail.example.com" and "mail.example.com CNAME smtp.example.com" will cause problems.

Problem? You SHOULD NOT have any other records pointing to a CNAME. At the very least, this causes unnecessary indirection (an extra step for looking up a domain name).