# Local DNS Lab

**35 Points**

**Due Date:Lesson 10**



## 1 Lab Overview

The purpose of this lab is to introduce you to the Domain Name System Protocol (DNS). DNS is the foundation on which the Internet operates. Without it, most everything we use today, from Active Directory services to simple web browsing and email services, would cease to function or function poorly. Disrupting DNS can lead to a denial of service condition. A poorly configured DNS can also reveal sensitive information about our networks. Further, subverting DNS can allow an adversary to control the resolution of domain names to IP addresses and thereby redirect traffic to destinations of his choice. Thus it is imperative to understand how DNS works and how to ensure a DNS implementation is reliable and secure.

You will explore DNS configuration options in both a LINUX environment and Windows Server environment. You are provided with an `Ubuntu` LINUX VM running BIND9 and a Windows 2008 Enterprise server VM. You will configure the BIND9 machine to serve as the master for your name domain providing resolution for external queries. The Windows 2008 Server will provide authoritative resolution for internal resources.

Your deliverables for this lab are your answers to the pre-lab questions, lab questions/task proof and lab summary IAW format on the course website. Submit one report per partner team. **This is a tough lab; understand the rubric, and start early. THESE SKILLS WILL BE SEEN AGAIN DURING THE CAPSTONE EXERCISE.**

## 2 Selected References

- Internet Engineering Task Force (IETF) Requests For Comment(RFC) http://www.ietf.org/rfc.html
  - RFC 1034, Domain Names - Concepts and Facilities
  - RFC 2181, Clarifications to the DNS Specification
  - RFC 3833, Threat Analysis of the Domain Name System
  - RFC 4033, DNS Security Introduction and Requirements
- DNS HOWTO: http://www.tldp.org/HOWTO/DNS-HOWTO.html
- BIND 9 Administrator ReferenceManual : http://www.bind9.net/manual/bind/9.3.2/Bv9ARM.ch01.html
- Microsoft Corporation. “DNS Server.” http://technet.microsoft.com/en-us/windowsserver/dd448607
- Kozierok, Charles M. “The TCP/IP Guide - TCP/IP Domain Name System (DNS)” http://www.tcpipguide.com/free/t_TCPIPDomainNameSystemDNS.htm

# 3 Pre-lab Questions

- **Q1.** What port(s) and transport layer protocol(s) does DNS use? What port(s) does the BIND controlling mechanism (RNDC) use?
- **Q2.** What is the difference between an iterative query request and a recursive query request?
- **Q3.** What are the following DNS records used for: A ; AAAA ; SOA ; NS ; MX; CNAME ?
- **Q4.** What is the difference between an authoritative DNS response and a non-authoritative response?
- **Q5.** What is a DNS zone transfer? What are zone transfers used for?
- **Q6.** Name a potential security vulnerability which currently exists in the DNS protocol. What is a way to mitigate that threat?
- **Q7.** What is DNSSEC? List one way DNSSEC makes DNS better?

# 4 Lab Environment

We need to setup the lab environment like the figure 1 below. You will configure at least three VMs: one `Ubuntu` (use "Observer") to act as the external server, one `Ubuntu` machine (use "victim") to act as a client and one Windows 2008 VM to act as an internal server. The lab description below uses one user machine, but you can also configure another host like `Ubuntu` attacker. The lab description will refer to the External DNS Server's as `10.172.xxx.4`[1] Later, we will incorporate the Windows server., internal as `10.172.xxx.100`, and user space as listed below. However, in your lab, you will use your IP addresses, fill in the list below for reference. Make it clear in your report which address is for which machine. The diagrams will refer to the third octet as "171", but they are consistent with respect to the other portions of the address.

- External DNS Server IP (`Ubuntu`) _____
- Internal DNS Server IP (W2K8) _____
- Internal Host IP(s) _____

Finally, to assist you in the lab write-up procedures, periodically take screenshots of what you are doing. Doing this as you go is MUCH easier than at the end.

**Note:** See CS482 online lab set-up page for any technical issues setting up the lab.

## 4.1 Install and configure the `Ubuntu` DNS server

**Step 1: Verify networking connectivity.** Before we begin ensure the following:

- the external server has connectivity to the Internet
- that all three hosts can ping each other.

[1] XXX represents the student's unique CS482 number.

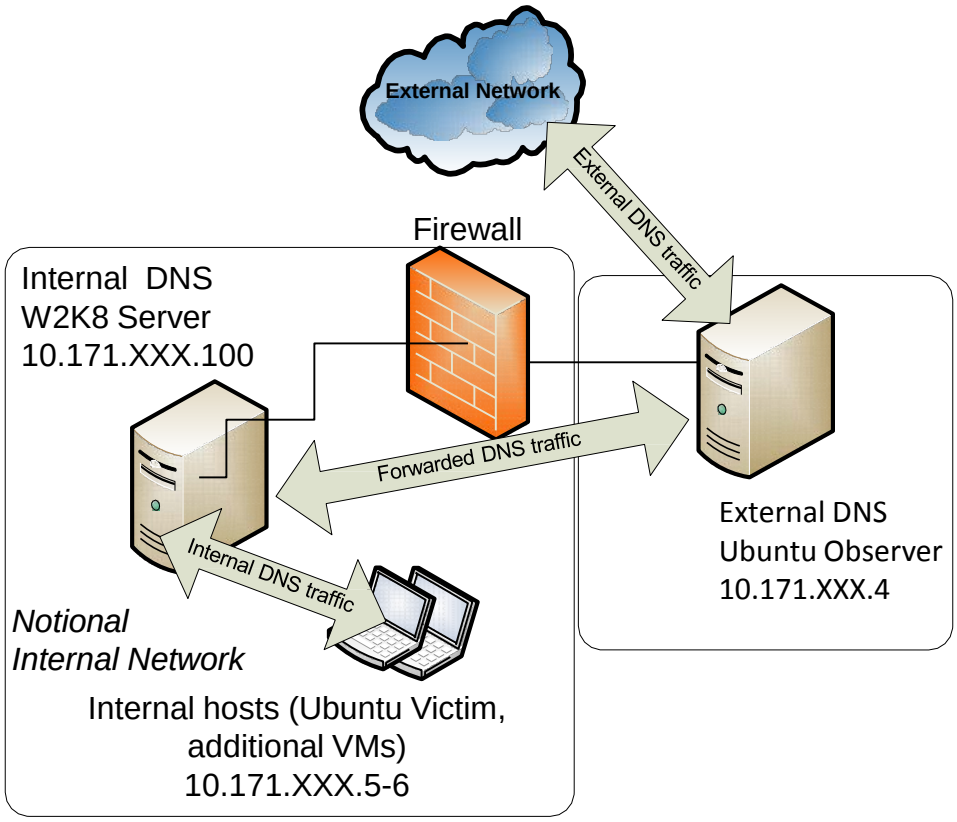


Figure 1: The Lab Environment Setup

If there is an issue, stop and fix it now! DNS will not work until you have fixed these layer 3 issues.

**Q8. Enter the command** `named -v` **on the Observer (External DNS)** 10.172.XXX.4. **What version of BIND are we running?**

Next, your external DNS server `Ubuntu` system (observer) must be configured to look at itself for name resolution instead of at another name server. To do this update, the `/etc/resolv.conf` file must be updated. Enter

`nameserver 10.172.XXX.4` and comment out (#) any existing name servers. NOTE: Changes to this file are temporary. To have change the namerserver permanently (i.e. to remain after your reboot your computer), edit `/etc/network/interfaces`.

To verify changes that will be made during the lab; there are two tools we will be using. Both are either included with the installation of BIND or are included with the operating system itself. The first tool, `nslookup`, is included with almost every operating system. The second tool, `dig`, is available by default on most UNIX or Linux operating systems and is available with any BIND installation (including the Windows binaries). To verify your configuration run the following command:

- `nslookup` www.cnn.com

If the result from this command is your name server's IP address (i.e. itself) then the resolver on external DNS server `Ubuntu` system (observer) is configured correctly.

**Step 2: Create the** `named.conf.options` **file.** The DNS server needs to read the `/etc/bind/named.conf` configuration file to start. This configuration file usually includes an option file called `/etc/bind/named.conf.options`. Please add the following content to the option file:

```
options {
    version "eecs";
    directory "/var/cache/bind/";
    dump-file "/var/cache/bind/dump.db";
    forwarders {
        8.8.8.8;
        };
    dnssec-validation auto;
    allow-query-cache {
        10.172.XXX.0/24;
        };
    allow-transfer {
        10.172.XXX.100;
        };
    allow-recursion {
        10.172.XXX.0/24;
        };
    auth-nxdomain no; # conform to RFC1035
    //listen-on-v6{ any; };
    listen-on{
        10.172.XXX.4;
        };
    //query-source address * port 5353;
};
```

- It should be noted that the file `/var/cache/bind/dump.db` is used to dump DNS server's cache.
- The options in the brackets can contain one or more values
- The `listen-on` option specifies the IP addresses on which your DNS server will respond. Edit this option to reflect the IP address of your `Ubuntu` VM, e.g. `10.172.XXX.4`. [Note: if you wished the server to only respond to queries from the local machine, not the network, you would set this to `127.0.0.1`]
- `version` changes the way we advertise our BIND version. We generally do not want to advertise the version we are running because an attacker can make use of that information. Feel free to change it; indulge your creativity in a SFW way. You do need to include the double quotes and semicolon.
- `allow-transfer` option that specifies which IP addresses can perform zone transfers with this server. We are going to put your Windows 2008 Server VM IP address in this field.

  **Q9. Why do we want to restrict zone transfers to this one IP address?**

- Add an `allow-recursion` option that specifies your subnet. This allows only hosts in your subnet to submit recursive queries to your DNS server. Also, add an `allow-query-cache` option that specified your subnet. Note: you can use CIDR notation inside the { }. For example, we could specify

a subnet as:
`{200.1.2.0/24;};`

**Q10. Why do we want to prevent those outside our network from being able to have our DNS server perform recursive domain name resolution or query?**

- `forwarders` option tells the server where to forward queries it cannot answer itself. Here it is set to the google name server `8.8.8.8`.
- The `query-source address * port NNNNN;` option allows you to specify a single port as the source for DNS queries sent from your server. Leave this option commented out. Although this might make it easier to work with a restrictive firewall, it also significantly reduces your resistance to cache poisoning.

**Q11. Why does limiting the source address to a single port make you more susceptible to DNS cache poisoning?**

**Step 3: Create zones.** Assume that we own a domain: `example.com`, which means that we are responsible for providing the definitive answer regarding `example.com`. Thus, we need to create a zone in the DNS server by adding the following contents to `/etc/bind/named.conf`. It should be noted that the `example.com` domain name is reserved for use in documentation, and is not owned by anybody, so it is safe to use it. Finally, enter the reserve look-up zone with the three octets and `in-addr.arpa` as written; the `in-addr.arpa` lets the service know that it is an IPv4 address.

```
zone "example.com" {
        type master;
        file "/var/cache/bind/example.com.db";
        allow-transfer {10.172.XXX.100;};
      };

zone "XXX.172.10.in-addr.arpa" {
        type master;
        file "/var/cache/bind/10.172.XXX";
        allow-transfer {10.172.XXX.100;};
      };
```

**Step 4: Setup zone files.** The file name after the `file` keyword in the above zones is called the zone file. The actual DNS resolution is put in the zone file. In the `/var/cache/bind/` directory, compose the following `example.com.db` zone file, i.e. `/var/cache/bind/example.com.db`

```
$TTL 3D
@       IN      SOA     ns.example.com. admin.example.com. (
        2008111001      ;serial, today's date + today's serial number
        8H              ;refresh, seconds
        2H              ;retry, seconds
        4W              ;expire, seconds
        1D)             ;minimum, seconds

@       IN      NS      ns.example.com. ;Address of name server
```

```
@       IN      MX      11      mail.example.com.
;Primary Mail Exchanger the "11" is pref field not a typo!

www     IN      A       10.172.XXX.5 ;Address of www.example.com
mail    IN      A       10.172.XXX.6 ;Address of mail.example.com
ns      IN      A       10.172.XXX.4 ;Address of ns.example.com
ns1     IN      A       10.172.XXX.4 ;Address of ns1.example.com
ns2     IN      A       10.172.XXX.4 ;Address of ns2.example.com
ns3     IN      A       10.172.XXX.4 ;Address of ns3.example.com
ns4     IN      A       10.172.XXX.100 ;Address of ns4.example.com
ns5     IN      A       10.172.XXX.100 ;Address of ns5.example.com
*.example.com. IN A     10.172.XXX.4  ;Address for other URL in
                                        ;example.com. domain
```

The symbol '@' is a special notation meaning the origin from the `named.conf`. Therefore, '@' here stands for `example.com`. 'IN' means internet. 'SOA' is short for Start Of Authority. This zone file contains multiple resource records (RRs): a SOA (Start Of Authority) RR, a NS (Name Server) RR, a MX (Mail eXchanger) RR, and 5 A (host Address) RRs.

We also need to setup the DNS reverse lookup file. In the directory `/var/cache/bind/`, compose a reverse DNS lookup file called `10.172.XXX` for `example.com` domain:

```
$TTL 3D
@       IN      SOA     ns.example.com. admin.example.com. (
                2008111001
                8H
                2H
                4W
                1D)
@       IN      NS      ns.example.com.

5       IN      PTR     www.example.com.
6       IN      PTR     mail.example.com.
4       IN      PTR     ns.example.com.
4       IN      PTR     ns1.example.com.
4       IN      PTR     ns2.example.com.
4       IN      PTR     ns3.example.com.
100     IN      PTR     ns4.example.com.
100     IN      PTR     ns5.example.com.
```

**Step 5: Start a DNS server.** Now we are ready to start the DNS server. Run the following command:

```
   % sudo /etc/init.d/bind9 restart
or
   % sudo service bind9 restart
```

At this point you should see the service running when you issue the `netstat` command:

```
   netstat -anotlup | grep 53
```

You may also want to check the `syslog` to see if there any issues:

```
tail -f /var/log/syslog
```

If you have any issues see the "Troubleshooting section below."

### 4.2 Temporarily Configure the User `Ubuntu` Client (Victim) to the External DNS `Ubuntu` Server (Observer)

For task 4.3 on the user machines, we will configure the client to let the machine 10.172.XXX.4 be the default DNS server. We achieve this by changing the DNS setting file `/etc/resolv.conf` of the user machine:

```
nameserver 10.172.XXX.4 # the ip of the DNS server you just setup
```

Note: make sure this is the only nameserver entry in your `/etc/resolv.conf`.

When you get to the Windows 2K8 task, you will change the nameserver to the W2K8 IP.

### 4.3 Testing

After you have set up the lab environment according to the above steps, your DNS server is ready to be tested.

**Q12. Test your `Ubuntu` DNS server by running the following commands from the user `Ubuntu`– document your results:**
`dig ns1.example.com , dig mail.example.com , dig ns2.example.com,`
`and dig www.example.com`
Ensure the 'ANSWER' section gives you the correct IP address for each fully qualified domain name (FQDN).

You should be able to see something similiar to this:

```
<<>> DiG 9.5.0b2 <<>> www.example.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27136
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;www.example.com. IN A

;; ANSWER SECTION:
www.example.com. 259200 IN A 10.172.XXX.5

;; AUTHORITY SECTION:
example.com. 259200 IN NS ns.example.com.

;; ADDITIONAL SECTION:
ns.example.com. 259200 IN A 10.172.XXX.4

;; Query time: 80 msec
```

```
;; SERVER: 10.172.XXX.4#53(10.172.XXX.4)
;; WHEN: Tue Nov 11 15:26:32 2008
;; MSG SIZE  rcvd: 82
```

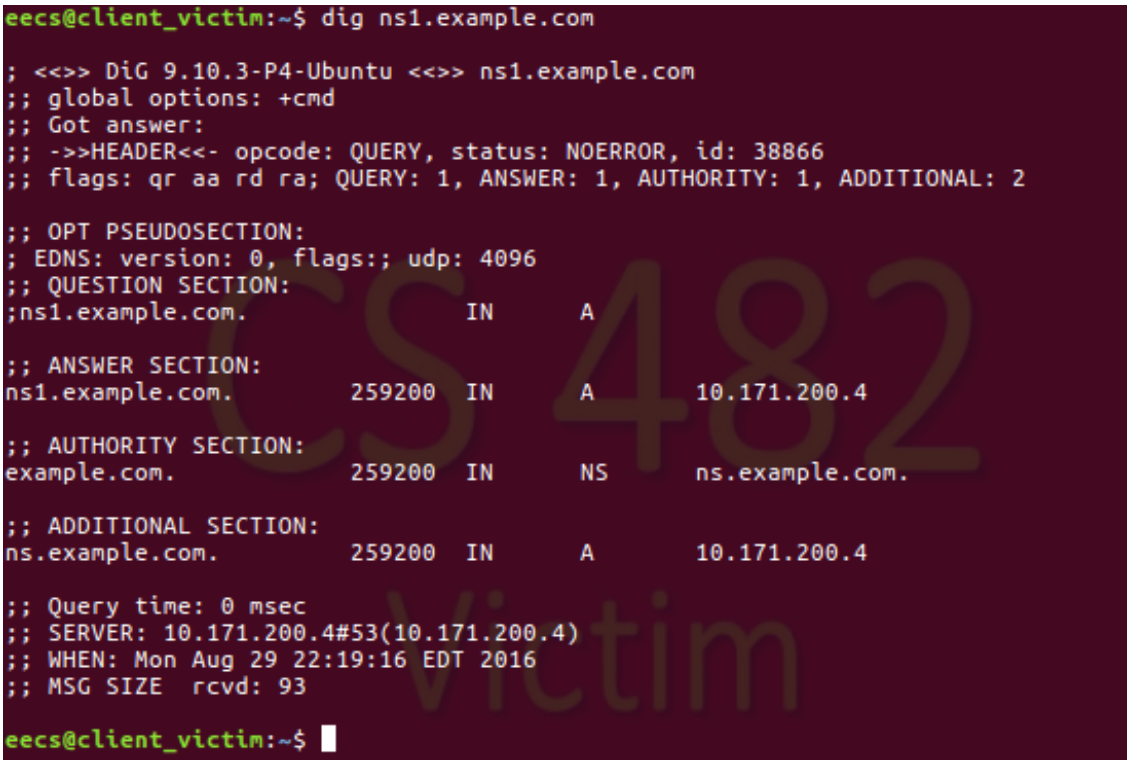


Figure 2: DiG command output

You can add -x to have dig do reverse lookups: e.g.

```
dig -x 10.172.XXX.y
```

**Q13. 'Dig' two IP addresses and ensure the 'ANSWER' comes back with the appropriate *.example.com names; document your results.**

## 4.4 Troubleshooting

If you have edited the files, but you get a failure or you are not able to resolve anything, remember your troubleshooting steps from IT350. First, confirm configurations! Take a hard look at the documents you created and ensure you have the right "." or space or ip address where they should be. Look for little misspellings. Errors messages can help you narrow the search. Try the following command right after your server fails to restart:

```
tail -f /var/log/syslog
```

This command should give you the last fews lines of errors in your syslog file, which should capture what went wrong and give you a hint as to what to fix. Often there will be a line number of the db file that has the issue! Second, if it is not clear what the issue is from tracing through the configs, it is time to confirm the network from layer 1 and up. Is layer 1 and layer 2 properly configured, i.e. are your interface cards correct

and connected to the virtual infrastructure? Do you have layer 3 connectivity? Can you ping end-to-end? A layer 7 protocol, like DNS, cannot work if the lower layers are broken! Often, one link in the chain is broken, causing issues.

### 4.5 Windows 2008 DNS and Master / Slave Zones

Because DNS servers are so critical to the infrastructure that supports the Internet and have a major impact on all network communications, we need to configure secondary servers to back up the master server in the event of a failure or some kind of attack. This configuration is known as a master and slave configuration. To make synchronization between the master and slaves easier, and to prevent errors, we will use what is known as *zone transfers* to copy the data from the master to the slaves.

We will be using the `Ubuntu` DNS server as the master for your `example.com` domain and the Windows 2008 Server will be the slave DNS server for that domain. **You must include Wireshark screenshots or printouts of the packet captures for the zone transfers you perform, so do not forget to start capturing packets on the** `Ubuntu` **server before you perform the tasks.**

- Log in to your W2K8Ent VM as Administrator (password = "@dm!n!str@t0r").
- Before we can install the DNS service, we must give the Windows 2008 Virtual Machine a static IP configuration: Assign the W2K8 VM the `.100` host IP address in your subnet. Add the netmask and gateway for your subnet. Again, verify layer 3 connectivity via a ping to each `Ubuntu` host.
- The DNS settings should be as follows:
  1. Preferred DNS Server = the IP address being assigned to your Windows 2008 VM
  2. Alternate DNS Server = the IP address of your `Ubuntu` VM.
- Next, we need to install the DNS server role on the Windows 2008 VM. **This action will not work if you do not set the IP address in the Virtual Machine first. Do not skip step 1.** Go to the W2K8 VM's Server Manager. Select 'add roles' and then 'DNS Server'. Click through and install the DNS service.
- When installation is complete, start the DNS Manager. Go Start → Administrative tools → DNS to start the DNS Manager. Expand the 'DNS'tree in the left pane and highlight your server (should be named something similar to `WIN-C5V8HOP`). Select the 'Action' menu and choose 'New Zone'. Select 'Secondary zone' , click next, the select 'Forward lookup zone', zone name: example.com.
- On the next page, specify the IP address of your primary DNS server (the `Ubuntu` VM). [Make sure WireShark is capturing, because the zone transfer should happen as soon as you enter the IP address . . . ] After the name is validated, click next.
- In the left pane you should now see your `example.com` zone under 'forward lookup zones'. If the zone transfer was successful you will see the same dns records you created on the `Ubuntu` VM. If you get a "Zone not transferred" error, press F5 a few times and give it time to refresh. If this doesn't work, go back and confirm your `Ubuntu` configuration.
- Now, go to the 'Action' menu and add another new secondary zone. This time create a reverse lookup zone. On the next page select 'IPv4 Reverse Lookup Zone'. On the next page select 'reverse lookup zone name' and enter the reverse zone name you specified on `Ubuntu` (`XXX.172.10.in-addr.arpa`) Click next and enter the primary DNS server IP address. Click through to finish. You should see the new zone under 'reverse lookup zones' in the left pane.

**Q14. Attach the WireShark packet capture screenshot or printout of the zone transfer**

**Q15. What transport layer protocol was used to make the zone transfer? Why?**

Use `nslookup` at the Windows command line to ensure `ns1.example.com , www.example.com, mail.example.com` resolve correctly.

Use `nslookup` to perform a zone transfer:

1. In the Windows 2008 VM, open a command shell
2. Type `nslookup` and press enter
3. Type `server 10.172.XXXY.4` for your External DNS `Ubuntu` server (Observer) IP address.
4. Type `ls -d example.com`, and press enter
5. You should see a listing of all of your hosts

**Q16. What change would you need to make to the BIND configuration to prevent all zone transfers?**

We are now going to start to configure the Windows 2008 DNS service to work in a classic split-dns mode. The `Ubuntu` VM, running BIND, will act as the external-facing DNS, answering queries from outside the local network. The Windows 2008 Server will provide the DNS service for internal hosts and only for those hosts. In addition, in an operational network, the internal server would actually maintain a seperate zone file that would segment the interal hosts from the external (we will not do this here). Therefore, the outside world would only be able to resolve the domain names in the external zone. The Windows server will forward any DNS queries it cannot answer to the `Ubuntu` server for external name resolution. We will emulate the diagram below; however, we do not have the firewall and separate subnets available on your virtualized environment.

**Q17. Briefly describe an advantage of using such a split-dns configuration. Think like an attacker!**

In DNS Manager, right-click on your example.com forward lookup zone and choose 'properties'. Change the type to 'primary zone'. Change your reverse lookup zone to primary as well. The internal DNS server will no longer pull zone transfers from the `Ubuntu` server.

Now, right-click on your DNS server in DNS Manager and choose properties. Then go to the forwarders tab. Add a forwarder to your `Ubuntu` server IP address.

At this point you would add DNS entries for internal hosts to the Windows 2008 Server only. These host names and IP addresses would not be available outside of the local network. Configure client `Ubuntu` to pull DNS services from the Windows server.

**Q18. Perform a foward and reverse dig from the client; document your results.**

# 5 Submission requirements

## 5.1 Partner Submission

Provide one written lab report, answering each question properly labelled with the number and original question, per partner team.

## 5.2 Individual Submission

Each member needs to submit a detailed lab reflection. This includes

- any challenging points or thoughts on what you found interesting during the lab
- time spent you personally spent and how much effort you put forth
- time your partner spent, and how much effort they put forth
- be sure document any external resources used.
- approximately one half page that explains a DNS attack mentioned in class or from http://www.networkworld.com/article/2886283/security0/top-10-dns-attacks-likely-to-infiltrate-your-network.htmlslide11.
  Be sure to highlight principles where appropriate.

## 5.3 Rubric

1. 7 pts for pre-lab question (1 pt per question)
2. 20 pts for lab tasks
   (a) 20 pts all tasks well-documented to include both zone transfers
   (b) 17 pts all tasks done but poorly-documented
   (c) 16 pts all `Ubuntu` tasks well-documented
   (d) 13 pts all `Ubuntu` tasks but poorly-documented
   (e) 6 pts partial implementation of `Ubuntu` server
   (f) 0 no tasks complete
3. 8 pts for individual submission

Again good documentation:

1. clearly enumerates tasks with a description of you did and evidence.
2. shows the progress you were able to achieve.
3. explains your troubleshooting attempts.
4. accurately describes an issue and the potential solution (if really good, I will give near full credit).