### Preamble

You must read the sections before Parts 1 – 4 before turning on your computers. Failure to start properly will result in running incorrect software and being unable to complete the lab.

Do not start your machine until you have read the ***Procedure*** section below.

### Environment

We are again using the virtual environment provided by VMware Workstation to set up a virtual network of computers. The Ubuntu machine UbuntuServer will host the server for DHCP, DNS and now Apache2 and two of the Windows machines will be used to test the server.

### Aims

Lan Segment 1

WINPE2

UbuntuServer

UbuntuServer communicates with outside world on eth0 interface via NAT on the host PC

Internet

Host PC

VMWare environment

WINPE1

### To install and configure an Apache webserver to host two websites on a single server and to secure those websites by using passwords and IP address blocking.

### Tools

Sudo will give you elevated privileges to run commands. On Ubuntu, these privileges are highly restricted. You may need to invoke elevated privileges by typing sudo ifconfig, this will allow access to sudo for a short while. You will probably need to do this every few minutes during the lab. A painful work-around, but it does work.

Ipconfig (on Windows) and ifconfig (on Linux) will show us the network configuration on our VMs. On the Windows machines we can also control the acquisition and renewal of IP addresses and display other network information.

Checking Syslog is very useful to help you understand why things are not working. Print out the last part of the syslog file with the following to see why something is not working…

**sudo tail /var/log/syslog**

Check out the tail command on the man pages. The man pages are a useful source of information on how things work, for instance.

Web pages

<http://httpd.apache.org/docs/2.4/>This is the home\main page for Apache 2.4 installation and configuration.

<http://httpd.apache.org/docs/2.4/howto/auth.html> has documentation on how to perform authentication and authorisation. Authentication is any process by which you verify that someone is who they claim they are. Authorization is any process by which someone is allowed to be where they want to go, or to have information that they want to have.

<http://manpages.ubuntu.com/manpages/utopic/man8/a2ensite.8.html> Adding a virtual host from the sites-available to the sites-enabled directory.

<https://help.ubuntu.com/lts/serverguide/httpd.html> How to unload a website using a2dissite command to make changes become effective after a re-configuration, followed of course by using a2ensite command…

<http://httpd.apache.org/docs/2.4/logs.html> Information on what the log files can do for you.

<http://httpd.apache.org/docs/current/howto/access.html> Access control to the server.

Ping can be used to try to reach a host. Either an IP address can be used or the FQDN can be used. If the FQDN is used, then the resolver of the DNS is utilised to translate the FQDN into an IP address.

nslookup can be used to test the DNS server from one of the PCs or from the Ubuntu server. You will get helpful messages I either case.

dig is another DNS testing utility which can tell you if your DNS server is working.

Check out the *help.ubuntu.com* web pages for more information.

**Web browser** on WINPE1 and WINPE2 will be used to check the availability of the websites that you establish, http://[www.computing.org](http://www.ca304.org) and http://ns.computing.org .

### Procedure

1. Power on your computer and boot into the **Windows** tab.
2. Log onto the machine as usual with your student login/password.
3. Launch VMWare Workstation from the start menu.
4. We are working with the ususl virtual machines on this exercise, the UbuntuServer (a Linux VM) and two Windows PCs, named WINPE1 and WINPE2. Power up each of these machines in turn.
   1. If you are asked if you moved or copied any of the VMs, say “**I moved it**”.
   2. Cancel any other messages as they come by.
5. The UbuntuServer has an account set up on it with (all lower case)
   1. Username: student
   2. Password: **computing**
6. Make sure that your DHCP and DNS servers are working correctly, that the PCs are getting IP addresses and the address of the DNS server. Make sure that nslookup is resolving the IP addresses for UbuntuServer.computing.org, [www.computing.org](http://www.ca304.org), ns.computing.org and fixed.computing.org. If not then copy down the appropriate configuration files from loop.dcu.ie and install them. This should all be done before this lab commences.

### Marking

* 1 Marks: Configure the *www.computing.org* website, that is the default website.
* 1 Mark: Password protect the www.computing.org website
* 1 Mark: Establish another virtual host ns.computing.org
* 1 Mark: Allow access to ns.computing.org only from WINPE1 and deny access from WINPE2
* 1 Mark: Use the Apache logfiles for both sites to log and explain HTTP response codes for
  + 200
  + 404
  + 403
  + 401 (if present)
  + 304 (if present)

Part 0: Installing Apache software

To ensure that all installed packages are up to date as usual…

**sudo apt-get update**

Install Apache2 with the following command…

**sudo apt-get install apache2**

There are some utilities that you need also…

**sudo apt-get install apache2-utils**

# Apache2 logs

Any access problems or start-up errors in relation to your Apache2 configuration will be logged to the Apache2 logs. Use tail in a new Terminal window or Terminal tab:

* tail /var/log/apache2/access.log
* tail /var/log/apache2/error.log

These are **REALLY USEFUL** when unexpected things happen. Make sure you can see them, test these commands after the initial installation, before the unexpected happens.

Part 1: Setup of Default WWW Website

Apache2 version 2.4 is the current version of the Apache web server. The **main configuration file** for Apache2 is apache2.conf . It contains settings that are global to Apache2.

Apache2 ships with a virtual-host-friendly default configuration. That is, it is configured with a single default virtual host (using the VirtualHost directive), which can modified or used as-is if you have a single site, or used as a template for additional virtual hosts if you have multiple sites. If left alone, the default virtual host will serve as your default site, or the site users will see if the URL they enter does not match the ServerName directive of any of your custom sites. To modify the default virtual host, edit the file…

/etc/apache2/sites-available/000-default.conf

**sudo gedit /etc/apache2/sites-available/000-default.conf &**

The default virtual website in Apache should load once you have installed Apache2 and loaded localhost. Try it, if the Apache2 welcome page loads, the installation worked.

In order to allow the Apache2 server to resolve URLs and use the DNS, you need to turn on DNS lookups in the main Apache2 configuration file :

* sudo gedit /etc/apache2/apache2.conf
* HostnameLookups On

Reload the Apache2 service:

That’s it for the apache2.conf file.

**sudo service apache2 reload**

Now Apache can resolve URLs from the DNS server that you established from the previous lab.

The default website now needs to be reconfigured to load up your own HTML code instead of the default web page. To do this you need to edit the default virtual host file. Before you do, **make a copy of it** so that you can use that copy to create new virtual host files for ns and you can create new copies for any other virtual sites you want to be hosted on the same machine..

**sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/000-default.conf.old**

Now edit the default site configuration file

**sudo gedit /etc/apache2/sites-available/000-default.conf &**

Add the directives

* ServerName www.computing.org
* ServerAdmin studentname@list.dcu.ie
* DocumentRoot /var/www/html

These directives set up the following

* FQDN for the website
* the email address of the Webmaster
* the location of the default index.html file for the home-page of the website.

Copy the old /var/www/html/index.html to a backup name

**sudo cp /var/www/html/index.html /var/www/html/index.html.old**

Throw nothing away!

Now edit the index.html file for www.computing.org in /var/www/html/index.html with minimal HTML code to show that you are indeed accessing the correct website. A brief welcome message will suffice, something like “Hi I’m the WWW website”.

**sudo gedit /var/www/html/index.html &**

You should already have created a CNAME for www in the last lab for the DNS.

Test this website on the UbuntuServer with <http://localhost> it should now give you your own webpage and not the default webpage.

Test the website from both WINPE1 and WINPE2. You should see your default web page for [www.computing.org](http://www.ca304.org) on both machines.

Try generating a 404 response from the Windows machines by typing in a non-existent URL like XXX.computing.burp. Check your error logs as setup in the default .conf file. There are marks going for getting this right, so get it done at this stage. See page 4 above.

Part 2: Apache Authentication

You can now protect your new website with password authentication.

See the manual pages at <http://httpd.apache.org/docs/2.4/howto/auth.html>

To do this you need to establish a password file and then associate it with the website that you want to protect. To do this you use a utility called  *htpasswd*.

Visit the Ubuntu help webpage listed above to assist you…

You may need to create the path /usr/local/apache/passwd/ as a location to store the password file.

**sudo htpasswd -c /usr/local/apache/passwd/passwords brian**

The –c directive will create the file, so use it once only. You probably need to create this directory structure using mkdir.

Many username/password combinations can be created. Create at least this combination…

* Username: brian
* Password: stone

Passwords are case sensitive.

You also need to add a Directory directive to tell Apache how to control files for this website. Add the <Directory /var/www/html> directive in the 000-default.conf file to link this password file to the [www.computing.org](http://www.ca304.org) website. **Check the *help.ubuntu* website** mentioned above for advice on how to do this and specify the following

* AuthType <you-choose>
* AuthName “Replace this and put your message here”
* AuthUserFile <supply full path to password file>
* Require valid-user (see the online help)

This allows only users with valid username/password combinations.

Remember to reload the Apache2 service to enable the password protection.

Test this from both Windows machines. Use a wrong password first to generate an error in the log-files.

Part 3: Second Virtual Host

You will need to create another virtual host (apart from the default) now, called ns.computing.org.

Remember, you may need to **add a** **CNAME entry to your DNS** configuration file to resolve the URL for ns.computing.org. Go back to last week’s work and edit the file computing.org.db to include an extra definition along with the old www. Remember to restart bind9.

For Apache you must create a new “.conf” file for the new virtual host from the old default .conf file..

**sudo cp /etc/apache2/sites-available/000-default.conf.old /etc/apache2/sites-available/ns.conf**

Now edit this file to tell the Apache2 server how to configure this new site. The default ServerName directive for the default virtual site (WWW) was www.computing.org, but now we need a new website called ns.computing.org. To achieve this we give directives for

* ServerName (our new virtual webserver)
* ServerAdmin (optional email address, if not included then the email is the default from before)
* DocumentRoot (the location of the index.html and other files for this new website)

We could and should also create new logs for this site.

The DocumentRoot directive should direct the server to look for the HTML files under the /var/www directory, i.e. in /var/www/ns/public\_html directory as this is a more secure place for these files and provided for a better security model. Make sure that the correct permissions are attached to these directories and files.

Now enable the new ns website…

**sudo a2ensite ns**

**sudo service apache2 restart**

In this way we have set up a new Virtual Host with its own specific .conf file and patched it into the Apache2 server using this a2ensite utility by creating a link between sites-available and the sites-enabled directory entries.

You should now test access to the ns.computing.org website from both Windows machines.

Part 4: Apache Access Control

Only allow access to the ns.computing.org website from WINPE1 and deny access from WINPE2, based on the IP addresses. You specify this in the ns.conf file with a

<Directory /full-path-name>

directive, similar to what you did for authentication. Check the Ubuntu Help and Apache website for details.

http://httpd.apache.org/docs/current/howto/access.html

This new directive tells the Apache2 server how to handle the directories particular to this virtual host, i.e. what machines have access to it. This is Access Control as opposed to Authentication, which we did in **Part 2**

To reload this information into the ns.computing.org website you must restart the Apache2 service on the server.

**Remember to unload the ns website, reload it and THEN restart the apache2 service.**

Part 5: Apache Log-Files

EDIT 2.2 - 2.4

Using gedit, bring up the Apache2 access log files from /var/log/apache2 as specified in your apache2.conf file for your two websites Check the website for assistance: [http://httpd.apache.org/docs/2.4/logs.html](http://httpd.apache.org/docs/2.2/logs.html)

It may be advantageous to turn off line wrapping in gedit to allow you to see individual entries:

* Edit > Preferences > Uncheck ‘Enable text wrapping’

Cut and Paste evidence of the following HTTP Response Codes into a new file. You will need to explain these lines to the lab supervisors.

* 200
* 404
* 403
* 401 (if you have it)
* 304 (if you have it)

[THE END]