Unix for Telecommunications

Portfolio Task – P-Lab-01-Introduction

Pass Level Task

I. Introduction

In this lab you will learn about Unix like operating systems, in particular the FreeBSD operating system (http://www.freebsd.org/). You will be covering some basic system navigation and configuration tasks that you will need for future lab sessions. Labs are completed using the Remote Unix Lab Environment (RULE) accessed from a PC in a Swinburne lab or from a home computer.

Additionally the lab component will cover reconfiguring your shell environment, modifying your ssh server configuration, and configuring your system to support remote X forwarding.

II. Purpose

To gain and/or enhance the following practical skills:

- Introduction to Unix common commands
- Understanding the general principles of navigation and operation of the Unix command line environment
- Modifying the shell environment of a user
- Making changes to the SSH server configuration
- Enabling remote X forwarding

III. Preparation

You can prepare for this lab by reading some of the FreeBSD documentation available at http://www.freebsd.org/. You should also review the basic concepts of:

- Unix common commands
- SSH
- Remote X forwarding

IV. Methodology

Your lab supervisor will provide you with login details to two (or more) RULE hosts. You will have full administator (`root`) access to these hosts for the duration of the semester, you should change your password(s) to one that is easier to remember. You must use your lowest numbered RULE host to complete your lab work. If your project requires the use of two RULE hosts, you must use the two highest numbered RULE hosts to complete your project. If your project requires the use of only one RULE host, you must use the second lowest (or highest if you have only been allocated two hosts) RULE host for your project.

A. Software

1) Open the `PuTTY` program and take a quick look at the available preferences in the opening dialog box. Since you will be spending most of your lab sessions in this program, you may want to explore and change some of its default settings. You will need the login details supplied to you by the lab demonstrator to SSH (secure shell) into your (*lowest numbered*) FreeBSD RULE host and login.
2) You should note that you can login to your RULE host using multiple concurrent sessions
3) Become familiar with copying and moving files to and from your RULE host “`student`” home directory using the windows program WinSCP.

B. Command syntax

1) The syntax of Unix command lines is `command <options(s)>`
2) Enter the command

```
date
```

Now enter the command

```
date "+DATE: %Y-%m-%d TIME: %H:%M:%S"
```

What is the output in each case?
3) Explore the following commands and become familiar with them:
   - The `ls` (list) command retrieves a listing of the contents of your current directory.
   - Run the `ls` command with the option "`-al`", eg. `ls -al`
   - The `pwd` (print working directory) command shows your current location in the file system.
   - You can move around using the `cd` (change directory) command. Eg. `cd /` (move to the root directory) or `cd ../` (move 'up' a directory) or `cd mydirectory` (move into mydirectory, assuming mydirectory folder exists in the correct directory). Before you move on, make sure you understand the difference between relative and absolute paths.
4) Use the `man` (manual) command with the option ls, eg. `man ls` . Look though the manual page to work out what lowercase 'a' and 'l' do to the ls command. The 'q' key exits a man page. Use the man command with some of the other commands you have already used. Eg. `man pwd` and `man date`.

C. File manipulation

1) Create a new directory in your home directory (`/home/student`) called "`test`" by using the command `mkdir test`. Be sure that you are in the '`student'` home directory before you try and create this directory by issuing and checking the output of the `pwd` command.

D. Text file editing

1) Inspect a text file with the program `cat`
   eg. `cat /usr/local/www/apache24/data/index.html`
2) Inspect a text file with the program `less`
   eg. `less /usr/local/www/apache24/data/index.html`
   Up and down arrows move through the file and 'q' key exits the program.
3) Create a text file called "`test.txt`" in your home directory (`/home/student/`) by issuing the command `ee test.txt` (after making sure you are in the right directory with `pwd`). Take some time to learn the basic features of the `ee` (easy edit) program.

E. Moving and deleting files

1) Delete a file from you home directory with the command `rm <filename>`
2) Delete a directory from your home directory with the `rm` command. What option did you have to use with the `rm` command to successfully remove the directory?
3) Rename a file in your home directory with the command `mv`

F. Output redirection

1) Issue the command `sysctl -a`?
   What happens?
   Issue the command `sysctl -a | less`
2) Issue the command `sysctl -a | grep "net"`
   What does this do?

3) Create two text files (`textfile1.txt` and `textfile2.txt`) with a small amount of content in each, issue the command `cat textfile1.txt >> textfile2.txt`
   Look at `textfile2.txt` what has happened here?
4) Now issue the command `cat textfile1 > textfile2.txt`
   What has happened this time?

G. Becoming the super user

1) Issue the command `su` to change to "super user" (or "administrator" in Widows terminology). You will be asked for your "`root`" password.
2) Edit the file "`motd`" (message of the day) in the `/etc` directory using the `ee` program. You should only be able to do this as the `root` user.
3) Exit `su` mode by typing `exit`. Exit your `student` user login session by typing `exit` again. Your PuTTY window will close. Login again to make sure your 'message of the day' has changed.
4) To log out of your RULE host type `exit` or `Ctrl^d` at a blank command line.

H. Changing the Unix shell

1) We will now change the `student` user from using the simple shell '`sh`' to the slightly better '`csh`'.
2) Issue the command `vipw` as the root user.
   - This will take you into the editing program `vi` – this can be difficult to use at first, in preference we would normally use `ee`.
   - Move the cursor to the very end of the line that reads (`.../home/student:/bin/sh`)
   - Press the '`i`' key to shift the program into insert mode
   - Change the end of the line to read – `.../home/user:/bin/csh`
   - Press `Esc` to exit insert mode, and then `:wq` to exit the editor and save the changes.
   - Restart your putty session. You should find that on your next login you have a different prompt indicating that you are now using `csh`.
3) Edit the text file (use `ee` rather than `vi`) `.cshrc` in your home directory – the `.cshrc` file is the configuration file for csh when run by the nominated user.
4) Why can you not see the `.cshrc` file when you run the `ls` command?
5) Under the line that reads (`alias ll ls -la`) add a new line that reads:
   `alias foo ls -lAh`
6) Re-login into your rule host again and issue the command `foo`. What is returned? Why?
7) Feel free to add any further alias you feel may be useful or rename the `foo` command to something that makes more sense.
8) If you don't like using `pwd` to constantly see which directory is the current directory, you may wish to search the Internet to work out how to add the current path to the shell prompt.

I. SSH

1) Try to log into your rule host as the user `root` directly using your `root` password. What does/doesn't happen?
2) Log in as `student` and run `su` to become `root`.
3) The configuration file for the ssh server (`sshd_config`) can be found in the directory `/etc/ssh`. We wish to edit this file.
4) There is a line in this configuration file that reads (`#Port 22`). What do you think editing this line would do?
5) Find the line that reads:
   `#PermitRootLogin no`
   and change it to read:

`PermitRootLogin yes`
What do you expect this to do? What does the '#' at the beginning of the line do?

6) Try to login to your rule host again as `root`. Does this work this time?
7) The `ssh` server needs to be restarted so it can re-read its configuration file. There are three ways to do this:
    - Kill the `ssh` server and then start it again. Read the manual pages (`man ps`) and (`man kill`) to determine how to locate a process ID for the `ssh` server and how to kill it. The ssh server can then be restarted using the command `sshd`. When locating the `ssh` program to terminate, you must kill the correct one.
    - Instead of actually stopping the server, you can send it the HUP signal to tell the server to reload its configuration and restart itself. How is this done using the kill command?
    - A better approach is to use the provided "`/etc/rc.d/sshd`" script to `stop/restart/start` any services your system provides. Given that your `ssh` server was automatically started using this approach, this is what you should use to restart the server
8) Restarting the `ssh` server on a remote computer is a risky proposition, if you made an error in your configuration the `ssh` server will not be able to restart and you will be unable to log back in. You should always test your changes by initiating a **second** login via `ssh` before exiting your current login, this allows you to fix the problem without losing access.

J. X and the GUI

1) Do some brief research on the X Windowing System.
2) Do some brief research on the **XMing** Windows Application.
3) Try executing the command `xcalc` on your RULE host. What happened?
4) We need to edit our `sshd_config` file again, this time changing the line that reads (`#X11UseLocalhost yes`) to read (`X11UseLocalhost no`).
5) Don't forget to restart the `ssh` server.
6) Launch the **XMing** application on your Windows computer.
7) Log back into your RULE host and try executing `xcalc` again.
8) Close the `xcalc` and this time execute the command `xeyes &`. What is the purpose of the `&`?
9) While `xeyes` is running, issue the command `ps -ax`
10) Where is `xeyes` doing it's processing? What about `xcalc`?

K. Automatically Starting SSH at System Boot

Note: *The following setup is already configured for your SSH server, however you will want to do this for other servers you configure during semester*

1) Once a service is properly configured and running, we often want to:
    - Start it automatically at boot-time, in case the system is rebooted for whatever reason
    - Enable starting/stopping/restarting the service via the use of the rc scripts – this simplifies the procedure and ensures that any dependent are also started if required
2) To configure both these tasks you need to edit the `/etc/rc.conf` file and add the line:

```
sshd_enable="YES"
```

3) You will now be able to start and stop SSH using the command:

```
/usr/local/etc/rc.d/sshd <start|stop>
```

V. Assessment

The due date for completion of this lab is via a demonstration to your lab supervisor **before** the end of the lab class in **Week 3**. Upon demonstration, your tutor will discuss your work with you in order to assess you on your competence

A. Completion of task in Doubtfire

Your tutor will assess you in class. In order for the submission to be marked as complete, you must discuss your work with the tutor.

Unix for Telecommunications

Portfolio Task – P-Lab-02-LaTeX
Pass Level Task

I. Introduction

In this lab you will explore the use of the LaTeX application to prepare documentation. LaTeX is a form of markup language for generating documents and is the traditional means of writing reports under Unix. LaTeX is also available for use under Windows and is free software.

II. Purpose

To gain and/or enhance the following practical skills:

- Generate a PDF document in **IEEE conference format** using LaTeX
- Understand how LaTeX works
- Include basic formatting elements using LaTeX
- Learn how to generate documents with LaTeX when provided a document class file

III. Preparation

You can prepare for this lab by reading some of the LaTeX documentation available at http://www.latex-project.org. A LaTeX compiler has already been installed onto your RULE system.

IV. Methodology

A. LaTeX Introduction

Examine the LaTeX web site listed above and explore the answers to the following questions:

1) Consider the possible advantages of using LaTeX to generate documentation over say a product such as Microsoft Word?
2) What is the purpose of the LaTeX preamble?
3) What are LaTeX packages and how are they included?
4) How do you generate bulleted/numbered lists?
5) How do you create tables under LaTeX ?
6) How do you insert images into a LaTeX document?
7) Why does the image move around the document under LaTeX s control?
8) How do you add a caption to a table/figure?

B. Generating Documents

1) Investigate the functionality of the programs (`latex`, `pslatex`, `pdflatex`, `dvips` and `ps2pdf`)
2) How do you go about generating final PDF documents using each of the three ***latex** commands

C. LaTeX Styles

1) Download the IEEE LaTeX template from https://www.ieee.org/conferences_events/conferences/publishing/templates.html (**Note:** You will need to use *IEEE conference template*, not the templates for journals.)
2) Put the templates into your LaTeX document source directory
3) How do you modify your document to generate using the IEEE style?
4) Where can you install the styles such that they are available to all users?
5) Did this previous step work? Why not?
6) What else do you have to do to make it work?

D. Labels, Referencing and Citation

1) What is the purpose of LaTeX `\label` command?
2) How does you automate references to tables/figures throughout your document ?
3) How does LaTeX manage references and citations?
4) What is the necessary LaTeX compile cycle when you add a new reference to your document?

V. Assessment

The due date for completion of practical work is **11:00pm**, exactly **six** days after your scheduled class.

Note: *The nominated submission day/time holds regardless of whether that day is a non-teaching day or public holiday*

To complete this lab you need to submit (to Doubtfire) an **IEEE conference** formatted PDF document generated by LaTeX that contains each of the following elements:

1) Title
2) Author name
3) Abstract
4) Section
5) Sub-section
6) Itemized list
7) Bullet list
8) Image/figure (including numbering and referencing)
9) Table (including numbering and referencing)
10) One reference cited in the text

A. Completion of task in Doubtfire

You will need to upload your LaTeX generated IEEE-formatted PDF document containing all the required elements to Doubtfire portfolio before the due date.

B. Tutor Discussion

In order for the submission to be marked as complete, you must discuss your work with the tutor.

Unix for Telecommunications

Portfolio Task – P-Lab-02-LaTeX
Distinction Level Task

I. Introduction

In this lab you will explore the use of the LaTeX application to prepare documentation. LaTeX is a form of markup language for generating documents and is the traditional means of writing reports under Unix. LaTeX is also available for use under Windows and is free software.

II. Purpose

To gain and/or enhance the following practical skills:

- Generate a PDF document in **ACM conference format** using LaTeX
- Understand the process of moving an existing LaTeX document to a different document class/template

III. Preparation

You should not attempt this task until you complete the **Pass** level task **P-Lab-02-LaTeX-P**.

IV. Methodology

Your task is to re-generate the document from the **Pass** level task **P-Lab-02-LaTeX-P** in the ACM conference format.

A. Generating document in ACM conference format

1) Download the ACM LaTeX template from https://www.acm.org/publications/proceedings-template (**Note:** You will need to use *ACM conference template*, not the templates for journals.)
2) Put the templates into your LaTeX document source directory
3) Do some research in how to modify your document to use the ACM style?

V. Assessment

The due date for completion of practical work is **11:00pm**, exactly **six** days after your scheduled class.

Note: *The nominated submission day/time holds regardless of whether that day is a non-teaching day or public holiday*

To complete this lab you need to submit (to Doubtfire) an **ACM conference** formatted PDF document generated by LaTeX that contains the exact same content as the **IEEE conference** formatted PDF generated for task **P-Lab-02-LaTeX-P**.

A. Completion of task in Doubtfire

You will need to upload your LaTeX generated ACM-formatted PDF document containing all the required elements to Doubtfire portfolio before the due date.

B. Tutor Discussion

In order for the submission to be marked as complete, you must discuss your work with the tutor.

Unix for Telecommunications

Portfolio Task – P-Lab-03-Bind
Pass Level Task

I. Introduction

In this lab you will configure your RULE host to become a DNS (Domain Name Service) server. The program that will be used to provide the DNS service is BIND (Berkeley Internet Name Domain), occasionally also referred to as `named`. BIND will be configured and run on your RULE host, then used by your RULE host to query and observe some interesting DNS behaviour. Once you have completed your lab, you will have created a new top level domain (`.unix`) that sits alongside the existing top-level domains (eg. `.com`, `.org`). This new domain will be available to anybody that uses your RULE host as a DNS server.

II. Purpose

To gain and/or enhance the following practical skills:

- Deploying and configuring BIND to provide DNS service
- Configuring BIND as a forwarding DNS server
- Creating a new top level domain and configuring forward and reverse zone files
- Understand issues relating to service configuration
- Process and respond to error messages in log files
- Configure services to auto-start in Unix

III. Preparation

You can prepare for this lab by reading some of the BIND documentation. You should also review the basic concepts of:

- What is the purpose and function of DNS
- What is DNS used for
- How hierarchical system functions
- What protocol/port numbers that DNS uses

IV. Methodology

A. Preliminary investigation

1) Explore the functionality of the `nslookup`, `host` and `dig` commands.
2) Issue the commands

```
nslookup www.swin.edu.au
host www.swin.edu.au
dig www.swin.edu.au
```

3) What is your output and what does it tell you?
4) What is the IP address of the Swinburne name server (DNS Server)?
5) Issue the command
   `nslookup 136.186.229.16`
6) What is returned? What does this tell you about the functionality of a DNS Server?
7) Issue the command
   `nslookup labbind.unix`
8) What is returned? Why? What does this mean?

B. Configuring the DNS Client

1) How does your RULE host know which DNS Server to use? (Hint: `man resolv.conf`)
2) What do each of the lines in this configuration file represent?
3) Changing the contents of this file is equivalent to doing what under Windows?
4) Configure your RULE host to use itself as a DNS server. (Hint: Use your rule hosts IP address as the DNS Server entry)
5) Do the previous `nslookup`'s you were asked to do work? Why/Why not?

C. Configuring the DNS Server

1) The BIND configuration information is stored in `/usr/local/etc/namedb/`. Examine the files in this directory.
2) Make a backup copy of the BIND configuration file (`named.conf`) Why do we do this?
3) Edit `named.conf` and make the following changes:
   - Comment out (add "`//`" to the beginning of the line) that reads

     ```
     listen-on { 127.0.0.1; };
     ```

   - Uncomment the section (remove the "`/*`" and "`*/`") that reads

     ```
     forwarders {
         127.0.0.1;
         };
     ```

   - Replace the `127.0.0.1` with the IP Address of one (or more) of Swinburne's DNS Servers (found earlier in the lab).
4) What is the purpose of the forwarders section of the `named.conf` file? (Hint: `man named.conf`)
5) At the very bottom of the file, under the section about zones, uncomment and remove/modify text so that it now reads

```
zone "unix" {
    type master;
    file "unix";
};

zone "230.186.136.in-addr.arpa" {
    type master;
    file "230.186.136.in-addr.arpa";
};
```

6) What is the purpose of the `zone` section, what does it represent in the DNS server?
7) Consider the *zones* section towards the end of the file. A number of zones are all configured to use the `empty.db` zone database file. What does this achieve?

D. Creating the DNS Database Files

1) Create the file `/usr/local/etc/namedb/working/unix` with the following contents (where **`xx`** is the number of your RULE host):

```
$TTL 86400

unix. IN SOA ns1.unix. admin.unix. (
                 2008070101 ; Serial
                 10800      ; Refresh
```

```
                  3600       ; Retry
                  604800     ; Expire
                  86400 )    ; Minimum TTL
; DNS Servers
@          IN   NS      ns1.unix.

; MX Record
@          IN   MX   1   ns1.unix.

; Machine Names
localhost  IN   A       127.0.0.1
ns1        IN   A       136.186.230.xx
labbind    IN   A       136.186.230.xx
labnmap    IN   A       136.186.230.21
```

2) Create the file /usr/local/etc/namedb/working/230.186.136.in-addr.arpa with the following contents (where **xx** is the number of your RULE host):

```
$TTL 3600

230.186.136.in-addr.arpa. IN SOA ns1.unix. admin.unix. (
                  2008070101 ; Serial
                  10800      ; Refresh
                  3600       ; Retry
                  604800     ; Expire
                  86400 )    ; Minimum TTL
; DNS Servers
@          IN   NS      ns1.unix.

; Machines
xx         IN   PTR     ns1.unix.
35         IN   PTR     labnmap.unix.
```

E. Starting BIND

1) Run `sockstat -4` and note the output.
2) Start BIND using the command `named`
3) Rerun `sockstat -4`. What does the output mean? What is the PID of your BIND server?
4) Test your DNS server by running (where **xx** is the number of your RULE host)

```
nslookup www.swin.edu.au
nslookup localhost.unix
nslookup ns1.unix
nslookup labbind.unix
nslookup labnmap.unix
nslookup 136.186.230.21
nslookup 136.186.230.xx
```

5) If some of these tests fail you will need to make changes to the configuration.
Note: *You **MUST** stop and restart named for it to use your new configuration settings. It is essential that you kill the running DNS server and start a new server running.*

F. Automatically Starting BIND at System Boot

1) Once a service is properly configured and running, we often want to:
   - Start it automatically at boot-time, in case the system is rebooted for whatever reason
   - Enable starting/stopping/restarting the service via the use of the rc scripts – this simplifies the procedure and ensures that any dependent servers are also started if required
2) To configure both these tasks you need to edit the `/etc/rc.conf` file and add the lines:

```
named_enable="YES"
named_chrootdir=""
```

   The second line is required as named cannot be run chrooted from within a jailed host environment (your rule host is a BSD jailed host)

3) Stop the currently running BIND service by issuing the command:

```
killall -9 named
```

4) You will now be able to start and stop BIND using the command:

```
/usr/local/etc/rc.d/named <start|stop>
```

V. Assessment

The due date for completion of practical work is **11:00pm**, exactly **six** days after your scheduled class.

Note: *The nominated submission day/time holds regardless of whether that day is a non-teaching day or public holiday*

A. Self Assessment

You can self-assess your progress at any time via the marking script available at http://ruleprimary1.caia.swin.edu.au

B. Completion of task in Doubtfire

Download the PDF output of the marking script from http://ruleprimary1.caia.swin.edu.au and submit it to Doubtfire. Your tutor will confirm completion of the lab by examing the rule marking log files on the rule server.

If you complete the task during class beforehand, you may demonstrate completion in class to your tutor.

Note: *The downloaded PDF is not evidence of successful completion of the lab, it is a document to demonstrate completion within your portfolio. Your tutor will assess the evidence via either direct confirmation via the marking script or via the log files generated when you run the marking script*

C. Tutor Discussion

In order for the submission to be marked as complete, you must discuss your work with the tutor

Unix for Telecommunications

Portfolio Task – P-Lab-04-Basic-Apache
Pass Level Task

I. Introduction

In this lab you will learn about the Apache web server application. You will perform some basic configuration of your RULE host to provide web services using Apache.

II. Purpose

To gain and/or enhance the following practical skills:

- Deploying and configuring Apache to provide a simple web service for a single website
- Understand issues relating to service configuration
- Process and respond to error messages in log files
- Configure services to auto-start in Unix

III. Preparation

You can prepare for this lab by reading some of the Apache documentation available at http://httpd.apache.org. You should also review the basic concepts of:

- TCP
- TCP Port Numbers
- Allocated/default TCP Port Number Allocation
- The HTTP Protocol

IV. Methodology

A. Apache Introduction

1) Apache has already been installed on your rule host, it would typically be installed under FreeBSD using ports. Locate the port installation directory under FreeBSD on your RULE host
2) Have a look at the `/etc/services` file. What do you think Unix uses this file for?

B. Apache Configuration File

1) Have a look at the Apache configuration file in `/usr/local/etc/apache24` (you do not have to understand all of the options in this file)
   - What port will Apache run on when started?
   - What directory from your RULE host will be served?
   - Locate where the error log and access log will be written to

C. Installing a Simple Web Page

1) Create some basic content for the Apache Server to serve. At a minimum this should include:
   - An `index.html` file
   - An image (eg. from http://www.freebsd.org/art.html)
   - Your name and student ID somewhere on the web page
2) Install your generated content into the directory you previously discovered would be served by Apache

D. Starting Apache

1) Run

```
sockstat −4
```

   and note the output
2) Start Apache using the command

```
apachectl −k start
```

3) Rerun

```
sockstat −4
```

   What does the output mean?
4) Access your new web site by browsing to your RULE host from one of the lab computers (Note: It is recommended not to use Internet Explorer due to it's caching behaviour)
5) Look at your `httpd-access.log` file
   - What is in this file?
   - What do each of the lines mean?
   - What does each field on each line represent?
6) Access a web page on the server (from the browser) that does not exist
   - Where can you find information about what went wrong?

E. Automatically Starting Apache at System Boot

1) Once a service is properly configured and running, we often want to:
   - Start it automatically at boot-time, in case the system is rebooted for whatever reason
   - Enable starting/stopping/restarting the service via the use of the rc scripts – this simplifies the procedure and ensures that any dependent are also started if required
2) To configure both these tasks you need to edit the `/etc/rc.conf` file and add the line:

```
apache24_enable="YES"
```

3) You will now be able to start and stop apache using the command:

```
/usr/local/etc/rc.d/apache24 <start|stop>
```

   Note: *You will have to stop the apache server using apachectl before using the rc scripts*

V. Assessment

The due date for completion of practical work is **11:00pm**, exactly **six** days after your scheduled class.

Note: *The nominated submission day/time holds regardless of whether that day is a non-teaching day or public holiday*

A. Self Assessment

You can self-assess your progress at any time via the marking script available at http://ruleprimary1.caia.swin.edu.au

B. Completion of task in Doubtfire

Download the PDF output of the marking script from http://ruleprimary1.caia.swin.edu.au and submit it to Doubtfire. Your tutor will confirm completion of the lab by examing the rule marking log files on the rule server.

If you complete the task during class beforehand, you may demonstrate completion in class to your tutor.

Note: *The downloaded PDF is not evidence of successful completion of the lab, it is a document to demonstrate completion within your portfolio. Your tutor will assess the evidence via either direct confirmation via the marking script or via the log files generated when you run the marking script*

C. Tutor Discussion

In order for the submission to be marked as complete, you must discuss your work with the tutor

Unix for Telecommunications

Portfolio Task – P-Lab-04-Basic-Apache
Credit Level Task

I. Introduction

In this lab you will learn about the Apache web server application. You will perform some basic configuration of your RULE host to provide web services using Apache.

II. Purpose

To gain and/or enhance the following practical skills:

- Deploying and configuring Apache to provide password protected access to a segment of a single website
- Understand issues relating to service configuration
- Process and respond to error messages in log files
- Self-investigation skills to learn how to deploy new ideas

III. Preparation

You should not attempt this task until you complete the **Pass** level task **P-Lab-04-Basic-Apache**

IV. Methodology

A. *Prepare your private Web page*

The first step will be to create the part of the web site we will eventually deploy password protection to

1) Create a sub-directory in the Apache served directory (*where you put your web site*) called `private`
2) Create and install some more content in the private directory
   Hint: *It is helpful to make sure this is slightly different than your original web site*
3) Browse to the private web site to ensure that it functions
4) Ensure that when you are browsing to http://ruleXX.caia.swin.edu.au/private you are presented with a web page and not a directory index

B. *Researching Simple Authentication*

1) Use the web to research the contents of the `.htaccess` file. What are some of the things you can configure with `.htaccess`
   Hint: *a good place to start is http:// www.javascriptkit.com/ howto/htaccess.shtml*
2) Look up how to use the htpasswd command
   Hint: *a good place to start is the man page for* `htpasswd`
   Hint: *pay particular attention to the* `-c` *option*

C. *Setup Apache to use .htaccess*

Edit the section in `httpd.conf` that refers to serving your web site

1) Change the line that reads "`AllowOverride None`" to "`AllowOverride All`"
2) Restart Apache

Why did you need to restart Apache?

D. Create the password file

We will be installing password protection to the private site, we need to create a password file containing the authorised username/password using the `htpasswd` command. You need to investigate creating a password file with the following properties

1) Password file is **NOT** located in any web-accessible directory
2) Password file is called `.htpasswd`
3) Username is `ruleperson`
4) Password is `ruleperson`

Why is the password file not stored in either the root web directory or the private web directory?

E. Create the .htaccess file

Create a `.htaccess` file in the private directory which enables Basic authentication using the previously generated (`.htpasswd`) password file. It is your task to investigate what options need to be deployed in the `.htaccess` file to achive this

F. Testing

1) Browse to the private folder on your web site http://ruleXX.caia.swin.edu.au/private
2) If you are not asked to enter a username/password it may be because the website is cached from your initial testing. Reload the page

What have you achieved? – or should have achieved

V. Assessment

The due date for completion of practical work is **11:00pm**, exactly **six** days after your scheduled class.

Note: *The nominated submission day/time holds regardless of whether that day is a non-teaching day or public holiday*

A. Self Assessment

You can self-assess your progress at any time via the marking script available at
http://ruleprimary1.caia.swin.edu.au

B. Completion of task in Doubtfire

Download the PDF output of the marking script from http://ruleprimary1.caia.swin.edu.au and submit it to Doubtfire. Your tutor will confirm completion of the lab by examing the rule marking log files on the rule server.

If you complete the task during class beforehand, you may demonstrate completion in class to your tutor.

Note: *The downloaded PDF is not evidence of successful completion of the lab, it is a document to demonstrate completion within your portfolio. Your tutor will assess the evidence via either direct confirmation via the marking script or via the log files generated when you run the marking script*

C. Tutor Discussion

In order for the submission to be marked as complete, you must discuss your work with the tutor

Unix for Telecommunications

Portfolio Task – P-Lab-05-Advanced-Apache
Pass Level Task

I. Introduction

In this lab you will extend the skills you learnt from the **BIND** and **Basic Apache** Labs. You will be required to modify your DNS server and Apache configuration to support virtual hosts. You will then tell your lab computer to use your modified DNS server and browse to the specified virtual hosts.

II. Purpose

To gain and/or enhance the following practical skills:

- Configure and understand virtual hosting under Apache
- Apply DNS skills learnt from previous lab
- Understand issues relating to service configuration
- Process and respond to server access/error messages in log files

III. Preparation

You can prepare for this lab by reviewing some of the Apache documentation on virtual hosts and on CGI scripts. You should also review the corresponding lecture notes.

IV. Scenario

You have been employed to host a website on your RULE host. The client is a company called “Armitage Chemicals”. This site should be hosted at http://armitagechemicals.unix.

When you complete this lab you should be able to:

- Configure a computer within Swinburne to use your RULE host as the DNS server
- Browse to http://armitagechemicals.unix and receive its content

NOTE: *Setting the DNS server on your home Windows-based computer using the VPN to access your RULE host does not work due to a limitation of the Cisco Windows VPN Client. If you are using Windows, it is likely that you will need to complete this lab on-campus.*

V. Methodology

A. Apache

You will be modifying the Apache configuration from the previous lab to support virtual hosting. Note that this may break the functionality of the **P-Lab-04-Basic-Apache** tasks if they have not already been assessed

B. Install the Web Content

1) Copy the website contents (available on Doubtfire) to your RULE host
2) Research how to extract the **Armitage Chemicals** website from the provided gzip tarball
3) Extract and install the **Armitage Chemicals** website to the directory: `/usr/local/www/armitagechemicals.unix/`

C. DNS Configuration

1) Update your BIND configuration such that the URL `armitagechemicals.unix` resolves to the IP address of your RULE host
   Hint: Refer to the BIND Lab handout for help if you have forgotten
   Hint: Don't forget to restart the BIND server after you have made any changes
   Hint: Make sure you only have one instance of BIND running
2) Perform a DNS lookup on your RULE host to confirm your DNS settings for `armitagechemicals.unix`
3) Set your local Windows computer to use your RULE host as a DNS server
4) Perform the same lookup on your Windows computer to verify that the URL resolves to the IP address of your RULE host

D. Configuring the Armitage Chemicals Virtual Host

1) Modify the Apache `httpd.conf` file to include the virtual host configuration file
   Hint: The line is commented out somewhere near the end of the file
2) Virtual hosts are configured in the file `/usr/local/etc/apache24/extra/httpd-vhosts.conf`
3) Add a new virtual host for the **ServerName** `armitagechemicals.unix`
   Ensure you set your Swinburne email address as the administrator
   Create a custom access/error log file for the host
   - `/var/log/armitagechemicals.unix-access_log`
   - `/var/log/armitagechemicals.unix-error_log`

   Set the document root to the directory where the web site is installed
4) Restart the Apache server
   Hint: run `/usr/local/etc/rc.d/apache24 stop` followed by `/usr/local/etc/rc.d/apache24 start`
5) Use your web browser to browse to http://armitagechemicals.unix
   Hint: You may need to turn off the proxy in your web browser configuration
6) If successful, you should be able to see the Armitage Chemicals website

VI. Assessment

The due date for completion of practical work is **11:00pm**, exactly **six** days after your scheduled class.

Note: *The nominated submission day/time holds regardless of whether that day is a non-teaching day or public holiday*

A. Self Assessment

You can self-assess your progress at any time via the marking script available at http://ruleprimary1.caia.swin.edu.au

B. Completion of task in Doubtfire

Download the PDF output of the marking script from http://ruleprimary1.caia.swin.edu.au and submit it to Doubtfire. Your tutor will confirm completion of the lab by examing the rule marking log files on the rule server.

If you complete the task during class beforehand, you may demonstrate completion in class to your tutor.

Note: *The downloaded PDF is not evidence of successful completion of the lab, it is a document to demonstrate completion within your portfolio. Your tutor will assess the evidence via either direct confirmation via the marking script or via the log files generated when you run the marking script*

C. Tutor Discussion

In order for the submission to be marked as complete, you must discuss your work with the tutor.

Unix for Telecommunications

Portfolio Task – P-Lab-05-Advanced-Apache
Distinction Level Task

I. Introduction

In this lab you will extend the skills you learnt from the **BIND** and **Basic Apache** Labs. You will be required to modify your DNS server and Apache configuration to support virtual hosts. You will then tell your lab computer to use your modified DNS server and browse to the specified virtual hosts. Furthermore, you will learn how to serve and run Common Gateway Interface (CGI) files using Apache.

II. Purpose

To gain and/or enhance the following practical skills:

- Configure and understand virtual hosting under Apache
- Apply DNS skills learnt from previous lab
- Understand issues relating to service configuration
- Process and respond to server access/error messages in log files

III. Preparation

You should not attempt this task until you complete the **Pass** level task **P-Lab-04-Advanced-Apache-P**.

IV. Scenario

You have been employed to host a second website on your RULE host (following installation, configuration, and hosting of the Armitage Chemicals website at http://armitagechemicals.unix). You are now required to host a second website – "The Aristocrats Family Theatre Restaurant Chain"at http://aristocrats.unix. "The Aristocrats Family Theatre Restaurant Chain" website is not available yet and the company want you to install a temporary page that consists of a single CGI script that shows a count-down until their first restaurant opens. This CGI script is also provided.

When you complete this lab you should be able to:

- Configure a computer within Swinburne to use your RULE host as the DNS server
- Browse to either http://armitagechemicals.unix or http://aristocrats.unix and receive the respective websites
- Both sites are to be hosted on the same RULE host

NOTE: *Setting the DNS server on your home Windows-based computer using the VPN to access your RULE host does not work due to a limitation of the Cisco Windows VPN Client. If you are using Windows, it is likely that you will need to complete this lab on-campus.*

V. Methodology

A. Install the Web Content

1) Copy the website contents (available on Doubtfire) to your RULE host
2) Install the **Aristocrats** CGI script to the `/usr/local/www/aristocrats.unix/` directory
3) Modify the **index.cgi** script to nominate a proper future opening date for the restaurant

B. DNS Configuration

1) Update your BIND configuration such that the URL `aristocrats.unix` resolves to the IP address of your RULE host
   Hint: Refer to the BIND Lab handout for help if you have forgotten
   Hint: Don't forget to restart the BIND server after you have made any changes
   Hint: Make sure you only have one instance of BIND running
2) Perform a DNS lookup on your RULE host to confirm your DNS settings for `aristocrats.unix`
3) Set your local Windows computer to use your RULE host as a DNS server
4) Perform the same lookup on your Windows computer to verify that the URL resolves to the IP address of your RULE host

C. Configuring "The Aristocrats Family Theatre Restaurant Chain" Virtual Host

1) Add a new virtual host for the **ServerName** `aristocrats.unix` using your configuration for **Armitage Chemicals** as a guide (make sure you specify the correct server directory **and** a unique set of log files)
2) Restart Apache and try to browse to the new virtual host
   - Why didn't this work?
   - Check your log files
3) Research on what changes you need to make to the `aristocrats.unix` virtual host to enable the CGI script to function
4) If it still does not function correctly, check your log files again or ask your lab supervisor for assistance

VI. Assessment

The due date for completion of practical work is **11:00pm**, exactly **six** days after your scheduled class.

Note: *The nominated submission day/time holds regardless of whether that day is a non-teaching day or public holiday*

A. Self Assessment

You can self-assess your progress at any time via the marking script available at http://ruleprimary1.caia.swin.edu.au

B. Completion of task in Doubtfire

Download the PDF output of the marking script from http://ruleprimary1.caia.swin.edu.au and submit it to Doubtfire. Your tutor will confirm completion of the lab by examing the rule marking log files on the rule server.

If you complete the task during class beforehand, you may demonstrate completion in class to your tutor.

Note: *The downloaded PDF is not evidence of successful completion of the lab, it is a document to demonstrate completion within your portfolio. Your tutor will assess the evidence via either direct confirmation via the marking script or via the log files generated when you run the marking script*

C. Tutor Discussion

In order for the submission to be marked as complete, you must discuss your work with the tutor

Unix for Telecommunications

Portfolio Task – P-Lab-06-PCAP-Programming

Pass Level Task

I. Introduction

In this lab you will learn how to use the **libpcap** (or similar) library with either the C or Python programming languages to parse packets. You will develop a tool to parse pre-captured PCAP files.

II. Purpose

To gain and/or enhance the following practical skills:

- Build a customised packet parsing tool/program instead of relying on existing tools (e.g tcpdump, tshark)
- Develop a better understanding of packet capture/sniffing tools
- Able to understand software documentation/examples and apply them
- Develop a deeper understanding of network traffic behaviour

III. Preparation

You can prepare for this lab by reading some of the documentation and examples available at:

- http://www.tcpdump.org/manpages/pcap.3pcap.html
- http://www.tcpdump.org/pcap.html
- https://eecs.wsu.edu/~sshaikot/docs/lbpcap/libpcap-tutorial.pdf
- http://recursos.aldabaknocking.com/libpcapHakin9LuisMartinGarcia.pdf

And for Python

- https://dpkt.readthedocs.io/en/latest/
- https://dpkt.readthedocs.io/en/latest/api/index.html

You should also review the basic concepts of:

- TCP/IP packets
- Packet sizes / lengths
- Existing packet analysers, such as tcpdump and tshark

IV. Requirements

You are to develop a software tool to parse a PCAP file and display a summary of all the packets within the file, as well as a brief summary of the overall contents of the file. The output format is standardised to allow for automated assessment of your work.

A. Program Requirements

Your program must be placed in the following directory:

```
/home/student/pcap_lab
```

Your program must be able to be executed directly from the command line and take one command line parameter. The command line parameter is the name of the PCAP file to parse. Execution should be as per:

```
./mtcpdump filename.pcap
```

Your program should open the PCAP file specified, process the contents of the file, and then print the required information.

B. Extracting and Displaying Individual Packet Information

For each packet in the PCAP file, you are required to extract and print the following information:

- Packet timestamp
- Source IP address and port number
- Destination IP address and port number
- Protocol (TCP / UDP)
- Packet length
- Packet Time-to-live (TTL)

All packet information is to be displayed to the screen (**stdout**) in the following format:

```
[2018-07-25 01:27:13.400464] - 172.16.10.62:5001 -> 172.16.11.72:32903 (tcp, len=52, ttl=63)
[2018-07-25 01:27:13.402465] - 172.16.10.62:5001 -> 172.16.11.72:32903 (tcp, len=52, ttl=63)
[2018-07-25 01:27:13.408465] - 172.16.10.62:5001 -> 172.16.11.72:32903 (tcp, len=52, ttl=63)
[2018-07-25 01:27:13.417466] - 172.16.10.62:5001 -> 172.16.11.72:32903 (tcp, len=52, ttl=63)
[2018-07-25 01:27:13.417486] - 172.16.11.72:32903 -> 172.16.10.62:5001 (tcp, len=52, ttl=64)
```

To enable automated assessment of your work, it is imperative that your output matches the format exactly.

In order to complete this task you will need to consider:

- When processing the PCAP files what tasks do you need to perform to extract individual fields for display?
- How might you modify mtcpdump such that it can perform a live capture rather than processing a PCAP file?

C. Displaying Summary Information from the PCAP File

Once all packets in the PCAP file have been parsed, you are required to output a brief summary to the screen (**stdout**) of the PCAP file including a count of the number of packets, the total number of bytes captured, and the packet size. The format of this output must be as per the following example:

```
------------------- SUMMARY -------------------

Total packets: 158141
Total bytes (Kbytes): 151375
Average packet size (bytes): 980

-----------------------------------------------
```

To enable automated assessment of your work, it is imperative that your output matches the format exactly.

In order to complete this task you will need to consider what mechanisms will you employ to allow you to extract this data from the PCAP file?

D. PCAP files

Two PCAP files are provided with the task resources:

- **`test1.pcap`** - A simple PCAP file that containing seventeen packets from two distinct flows. This file is provided to make it easy for you to test and debug your program. You can easily open this file in Wireshark and manually review the correctness of your program
- **`test2.pcap`** - A more complex PCAP file that contains multiple flows. Once you are confident that your program works properly, you can test it with this file.

When you execute the marking script to verify your program, it will execute your program against both these files as well as a third – hidden – PCAP file to confirm that your program works as intended.

V. Choosing your Programming Language

You are free to choose whichever programming language you like to develop your software. Before making your selection, it is strongly recommended you consider the task and read the associated links for information.

A. C or C++

The gcc compiler has already been installed on your rule host along with the libpcap library. This means that you will not have to install any further software to complete the lab using C or C++

Within the task resources available on Doubtfire, you will also find a sample C program with two functions (**`print_packet_info()`** and **`print_summary()`**). These functions will print parameterised variables to exactly match the format required to pass the lab assessment. It is strongly recommended you use these functions to help ensure your output meets the requirements for assessment.

B. Python

Python has not been installed on your rule host. If you would like to complete your lab using Python, you will need to research how to install Python and any extra libraries required on your rule host before you begin.

Within the task resources available on Doubtfire, you will also find a sample Python program with two functions (**`print_packet_info()`** and **`print_summary()`**). These functions will print parameterised variables to exactly match the format required to pass the lab assessment. It is strongly recommended you use these functions to help ensure your output meets the requirements for assessment.

Note: *Developing your software in Python will be easier than using C, however the task of installing Python effectively makes the complexity of either approach equal*

Note: *If you are intending to also complete the Distinction Level task, it is recommended to use Python to significantly make your overall task easier*

C. Other Languages

No other programming language has been installed on your rule host. If you would like to use an alternate programming language, you will need to research how to install it and any required libraries before you begin. It is also your responsibility to develop code to correctly output information in the required format. Only C and Python example code will be provided.

D. Report

If you are preparing a lab report for this lab, please answer the question as to how and why you chose your ultimate programming language to complete the task.

VI. Assessment

The due date for completion of practical work is **11:00pm**, exactly **six** days after your scheduled class.

Note: *The nominated submission day/time holds regardless of whether that day is a non-teaching day or public holiday*

A. Self Assessment

You can self-assess your progress at any time via the marking script available at
http://ruleprimary1.caia.swin.edu.au

B. Completion of task in Doubtfire

Download the PDF output of the marking script from http://ruleprimary1.caia.swin.edu.au and submit it to Doubtfire. Your tutor will confirm completion of the lab by examing the rule marking log files on the rule server.

If you complete the task during class beforehand, you may demonstrate completion in class to your tutor.

Note: *The downloaded PDF is not evidence of successful completion of the lab, it is a document to demonstrate completion within your portfolio. Your tutor will assess the evidence via either direct confirmation via the marking script or via the log files generated when you run the marking script*

C. Tutor Discussion

In order for the submission to be marked as complete, you must discuss your work with the tutor

Unix for Telecommunications

Portfolio Task – P-Lab-06-PCAP-Programming

Distinction Level Task

I. Introduction

In this lab you will learn how to use the **libpcap** (or similar) library with either the C or Python programming languages to parse packets. You will develop a tool to parse pre-captured PCAP files.

II. Purpose

To gain and/or enhance the following practical skills:

- Build a customised packet parsing tool/program instead of relying on existing tools (e.g tcpdump, tshark)
- Develop a better understanding of packet capture/sniffing tools
- Able to understand software documentation/examples and apply them
- Develop a deeper understanding of network traffic behaviour

III. Preparation

You can prepare for this lab by reading some of the documentation and examples available at:

- http://www.tcpdump.org/manpages/pcap.3pcap.html
- http://www.tcpdump.org/pcap.html
- https://eecs.wsu.edu/~sshaikot/docs/lbpcap/libpcap-tutorial.pdf
- http://recursos.aldabaknocking.com/libpcapHakin9LuisMartinGarcia.pdf

And for Python

- https://dpkt.readthedocs.io/en/latest/
- https://dpkt.readthedocs.io/en/latest/api/index.html

You should also review the basic concepts of:

- TCP/IP packets
- Packet sizes / lengths
- Existing packet analysers, such as tcpdump and tshark

IV. Requirements

You are to develop a second software tool to parse a PCAP file and display a summary of flow information within that file. The output format is standardised to allow for automated assessment of your work.

A. *Program Requirements*

Your program must be placed in the following directory:

```
/home/student/pcap_lab
```

Your program must be able to be executed directly from the command line and take one command line parameter. The command line parameter is the name of the PCAP file to parse. Execution should be as per:

```
./tcpsummary filename.pcap
```

Your program should open the PCAP file specified, process the contents of the file, and then print the required information.

B. Extracting and Displaying Flow Information

For each unique flow in the PCAP file, you are required to extract and print the following information:

- Flow tuple (source IP/Port, destination IP/Port, protocol)
- Timestamp of the first packet in the flow
- Total packets within the flow
- Total bytes within the flow

All packet information is to be displayed to the screen (**stdout**) in the following format:

```
--------------- PER-FLOW INFO -----------------

Flow: 172.16.11.72:35192-172.16.10.62:5000
Start time: 2018-07-25 01:25:32.370918
Total packets: 50078
Total bytes: 75110020

Flow: 172.16.10.62:5000-172.16.11.72:35192
Start time: 2018-07-25 01:25:32.392489
Total packets: 28154
Total bytes: 1504588

Flow: 172.16.11.72:32903-172.16.10.62:5001
Start time: 2018-07-25 01:25:32.483847
Total packets: 51247
Total bytes: 76864608

Flow: 172.16.10.62:5001-172.16.11.72:32903
Start time: 2018-07-25 01:25:32.520527
Total packets: 28662
Total bytes: 1529660

-----------------------------------------------
```

To enable automated assessment of your work, it is imperative that your output matches the format exactly. It is also important the order that flow information is printed is consistent, you are required to sort your flows in chronological order based on the timestamp of the first packet in the flow.

To simplify your code, you should consider packets flowing in the reverse direction of the flow being part of a different flow and have their own summary (eg. in a http flow between `192.168.0.1:45678` and `192.168.0.2:80`, packets from `192.168.0.1:45678` to `192.168.0.2:80` form a different flow to those from `192.168.0.2:80` to `192.168.0.1:45678`)

In order to complete this task you will need to consider:

- How to store information for all flows for later output?
- How to correctly order the output of flow information?
- How to update flow information as each packet is processed?
- How much extra consideration to put into the design of your solution to complete this task?
- If you are preparing a lab report, also answer how you might modify your program such that bi-directional packets of the same flow are combined into a single flow

C. PCAP files

Please develop and test your program using the same two PCAP files provided with the **P-Lab-06-PCAP-Programming-P** task resources.

When you execute the marking script to verify your program, it will execute your program against against the same three PCAP files as for the **P-Lab-06-PCAP-Programming-P** Pass Level task, keeping the contents and output of the third PCAP file hidden.

V. Choosing your Programming Language

You are free to choose whichever programming language you like to develop your software. Before making your selection, it is strongly recommended you consider the task and read the associated links for information. Unlike the **Pass Level** task, completing this **Distinction Level** task will be easier using a higher level programming language like Python. Otherwise the same issues apply as per the Pass task with your choice of programming language.

A. Sample Programs

As with the **Pass Level** task, we have provided two sample programs (in both C and Python) within the task resources available on Doubtfire. Both these sample programs contain three functions (**`flow_summary_header()`**, **`flow_summary_header()`** and **`flow_summary_single()`**). These functions will print parameterised variables to exactly match the format required to pass the lab assessment. It is strongly recommended you use these functions to help ensure your output meets the requirements for assessment.

VI. Assessment

The due date for completion of practical work is **11:00pm**, exactly **six** days after your scheduled class.

Note: *The nominated submission day/time holds regardless of whether that day is a non-teaching day or public holiday*

A. Self Assessment

You can self-assess your progress at any time via the marking script available at
http://ruleprimary1.caia.swin.edu.au

B. Completion of task in Doubtfire

Download the PDF output of the marking script from http://ruleprimary1.caia.swin.edu.au and submit it to Doubtfire. Your tutor will confirm completion of the lab by examing the rule marking log files on the rule server.

If you complete the task during class beforehand, you may demonstrate completion in class to your tutor.

Note: *The downloaded PDF is not evidence of successful completion of the lab, it is a document to demonstrate completion within your portfolio. Your tutor will assess the evidence via either direct confirmation via the marking script or via the log files generated when you run the marking script*

C. Tutor Discussion

In order for the submission to be marked as complete, you must discuss your work with the tutor

Unix for Telecommunications

Portfolio Task – P-Lab-07-NMap

Pass Level Task

I. Introduction

In this lab you will use the program NMap to discover a range of network services running on remote machines and perform some simple network auditing tasks.

WARNING: Although port scanning is not illegal, it is considered bad practice and may be viewed as a prelude to an attack by the operators of those hosts. Do not scan hosts and ports outside of the ones defined in this lab handout using your RULE host. Note that if you do, your actions can be traced and you may face consequences. Remember that you are bound by Swinburne's IT policies and procedures while using Swinburne's IT infrastructure.

II. Purpose

To gain and/or enhance the following practical skills:

- Using NMap to discover a range of network services running and simple network auditing tasks
- Understand concepts and processes in determining which services are running on a remote system

III. Preparation

You can prepare for this lab by reading some of the NMap documentation available at https://nmap.org/.

IV. Methodology

A. Special Note

All commands in this lab should be run as user **`student`** rather than as **`root`**. Running NMap as **`root`** on the RULE host does not work properly due to limitations imposed by the virtualisation system deployed using BSD jails. If you are exploring NMap beyond the commands described in this lab you may receive messages such as "Operation not permitted". If this occurs it is because you have entered a command that required low-level **`root`** access to the underlying hardware. Using NMap at home on a real host will allow you to fully explore NMap. Again, it is considered poor practice to run NMap against hosts that you do not manage and may be considered as a prelude to an attack by the operators of those hosts. Please consider this when you run NMap from home.

B. NMap Exploration

1) NMap has already been installed on your rule host. If it would not be installed already, how could you install NMap under FreeBSD?
2) Read the NMap documentation, what is the purpose of running NMap in verbose mode? Why should you do this during the lab exercises?
3) Perform a "Host Discovery" on the subnet `136.186.230.0/24`. What does the result show?
4) What ports are included in a default port scan? Why?
5) Scan the following hosts and comment on what ports are open:
   - `jbut.caia.swin.edu.au`
   - Your other Rule host(s)
6) Repeat the scan on the above hosts but this time scan **all** ports rather than just the default ports.
7) Comment on why you think some machines might take longer to scan than others.

C. Probing a hosts network services in more detail

We will now explore the host `rule21.caia.swin.edu.au`

1) What network applications/services are running on this host?
2) What versions of these applications are running on this host?
3) How might this information be used for:
   - A hacker attempting to compromise the system
   - A network administrator attempting to protect the system
   - The server administrator

V. Assessment

The due date for completion of practical work is **11:00pm**, exactly **six** days after your scheduled class.

Note: *The nominated submission day/time holds regardless of whether that day is a non-teaching day or public holiday*

To complete this lab you need to submit (to Doubtfire) a simple (unformatted) PDF document containing a list of all services running on `rule21.caia.swin.edu.au`

Note: *A PDF upload is required as Doubtfire cannot currently accept uploads of text files*

A. Completion of task in Doubtfire

You will need to upload your PDF file containing the list of running services to your Doubtfire portfolio before the due date

B. Tutor Discussion

In order for the submission to be marked as complete, you must discuss your work with the tutor

Unix for Telecommunications

Portfolio Task – P-Lab-07-NMap

Credit Level Task

I. Introduction

In this lab you will use the program NMap to discover a range of network services running on remote machines and perform some simple network auditing tasks. You will also perform some investigative work to scan for hidden information.

WARNING: Although port scanning is not illegal, it is considered bad practice and may be viewed as a prelude to an attack by the operators of those hosts. Do not scan hosts and ports outside of the ones defined in this lab handout using your RULE host. Note that if you do, your actions can be traced and you may face consequences. Remember that you are bound by Swinburne's IT policies and procedures while using Swinburne's IT infrastructure.

II. Purpose

To gain and/or enhance the following practical skills:

- Using NMap to discover a range of network services running and simple network auditing tasks
- Understand concepts and processes in determining which services are running on a remote system
- Investigate a remote system to locate hidden information

III. Preparation

You should not attempt this task until you complete the **Pass** level task **P-Lab-04-NMap-P**.

IV. Methodology

A. Challenge investigation

1) The host `rule21.caia.swin.edu.au` is running a secret application. Using NMap and other network application software, can you determine what the secret message is?

V. Assessment

The due date for completion of practical work is **11:00pm**, exactly **six** days after your scheduled class.

Note: *The nominated submission day/time holds regardless of whether that day is a non-teaching day or public holiday*

To complete this lab you need to submit (to Doubtfire) a simple (unformatted) PDF document containing the secret message on `rule21.caia.swin.edu.au`

Note: *A PDF upload is required as Doubtfire cannot currently accept uploads of text files*

A. Completion of task in Doubtfire

You will need to upload your PDF file containing the secret message to your Doubtfire portfolio before the due date

B. Tutor Discussion

In order for the submission to be marked as complete, you must discuss your work with the tutor

Unix for Telecommunications

Portfolio Task – P-Lab-08-tcpdump
Pass Level Task

I. Introduction

In this lab you will use the program **tcpdump** to perform some ‘detective’ work on packets captured in the situation shown below.

A Unix for Telecommunications student (Lawrence Slowman) is sitting at the client machine. He connects to the server using a number of different services and performed a number of different tasks. Using the packet trace file captured on the ‘sniffing’ machine and the program **tcpdump** – it is your job to discover as much as possible about what Lawrence did during this session.

II. Purpose

To gain and/or enhance the following practical skills:

- Learn how to use **tcpdump** for traffic analysis
- Explore and use appropriate **tcpdump** options
- Understand the value and implications of **tcpdump**

III. Preparation

Download a copy of the packet trace file for examination from Doubtfire and place it on your rule host for examination prior to the lab.

Also you can read the **tcpdump** documentation at http://www.tcpdump.org

IV. Methodology

A. *tcpdump Introduction*

1) **tcpdump** has already been installed on your rule host, how would you go about installing **tcpdump** under FreeBSD?
2) After reading the **tcpdump** documentation, in what way is the name **tcpdump** misleading?

B. *Exploring the packet trace file*

1) Don’t forget to uncompress the packet trace file you have downloaded (*Hint: man gzip*)
2) You can examine the trace file as a regular user but live capture requires **root** privileges, why?
3) What do each of the fields before the ‘`:`’ in the output of **tcpdump** mean?
4) What are the IP addresses of the two hosts in the traffic trace? What are the hostnames assigned to these hosts?
5) How would you tell **tcpdump** to give you:
   a) Only the web traffic from the trace file
   b) Packet information in HEX format
   c) Packet information in ASCII format
   d) Full versus summarised packet information
6) How are these options useful in exploring traces using **tcpdump**?
7) What is the purpose of **tcpdump**’s `-n` option? When is it useful?

C. *What did Lawrence do?*

With your basic **tcpdump** skills, answer the following questions about the packet trace file:

1) Over what time frame was the trace file captured? (duration of session in *seconds*)
2) What four protocols did Lawrence use and in what order?
3) What web site did Lawrence visit that was hosted on `136.186.229.138`?
4) What happened the first time Lawrence tried to log in using **ftp**?
5) What file did Lawrence retrieve using **ftp**?
6) What commands were issued by Lawrence across the **ssh** session?
7) What was the contents of the email sent to Lawrence? Who sent it and when?
8) What is Lawrence's password?

V. Assessment

The due date for completion of practical work is **11:00pm**, exactly **six** days after your scheduled class.

Note: *The nominated submission day/time holds regardless of whether that day is a non-teaching day or public holiday*

To complete this lab you need to submit (to Doubtfire) a simple (unformatted) PDF document a brief answer to all the questions in Section IV-C.

Note: *A PDF upload is required as Doubtfire cannot currently accept uploads of text files*

A. *Completion of task in Doubtfire*

You will need to upload your PDF file containing the answers to your Doubtfire portfolio before the due date

B. *Tutor Discussion*

In order for the submission to be marked as complete, you must discuss your work with the tutor

Unix for Telecommunications

Portfolio Task – P-Lab-08-tcpdump
Credit Level Task

I. Introduction

In this lab you will use the program **tcpdump** to perform some 'detective' work on packets captured in the situation shown below.

A Unix for Telecommunications student (Lawrence Slowman) is sitting at the client machine. He connects to the server using a number of different services and performed a number of different tasks. Using the packet trace file captured on the 'sniffing' machine and the program **tcpdump** – it is your job to discover as much as possible about what Lawrence did during this session.

II. Purpose

To gain and/or enhance the following practical skills:

- Learn how to use **tcpdump** for traffic analysis
- Explore and use appropriate **tcpdump** options
- Understand the value and implications of **tcpdump**

III. Preparation

You should not attempt this task until you complete the **Pass** level task **P-Lab-04-tcpdump-P**.

IV. Methodology

After analysing Lawrence's activities in task **P-Lab-04-tcpdump-P**, you are now required to graph the network traffic pattern generated over time.

A. Generating Graphical Output

1) Explore the functionality of the commands **cut**, **tr** and **grep**
2) Use the above commands (and possibly others) in combination with **tcpdump**, pipes and redirection to produce a text file consisting of three columns, the first showing flow ID (IP addresses and port numbers), the second showing time in seconds+milliseconds and the third showing packet size for each packet sent from the server to Lawrence
3) Open this document in Excel or OpenOffice (or other graphing software), split the data into multiple groups, one for set of flows mapping to the four protocols used by Lawrence
   Hint: *Put one set of flow data to each worksheet*
4) Generate a new column containing cumultative bytes transferred against each time stamp
5) Plot graphs of *time (x-axis) versus cummulative bytes (y-axis) transferred* for each flow ID on each group of flows
   Hint: *You should have four graphs, one for each protocol used by Lawrence*
   Hint: *Each graph may have more than one unique flow*
   Hint: *Each flow within a graph should be plotted with a different colour – colour defines flow tuple*

V. Assessment

The due date for completion of practical work is **11:00pm**, exactly **six** days after your scheduled class.

Note: *The nominated submission day/time holds regardless of whether that day is a non-teaching day or public holiday*

To complete this lab you need to submit (to Doubtfire) a simple (unformatted) PDF document containing the graph generated in Section IV-A.

Note: *A PDF upload is required as Doubtfire cannot currently accept uploads of image files*

A. Completion of task in Doubtfire

You will need to upload your PDF file containing your graph to your Doubtfire portfolio before the due date

B. Tutor Discussion

In order for the submission to be marked as complete, you must discuss your work with the tutor

Unix for Telecommunications

Portfolio Task – P-Lab-08-tcpdump
Distinction Level Task

I. Introduction

In this lab you will use the program **tcpdump** to perform some 'detective' work on packets captured in the situation shown below.

A Unix for Telecommunications student (Lawrence Slowman) is sitting at the client machine. He connects to the server using a number of different services and performed a number of different tasks. Using the packet trace file captured on the 'sniffing' machine and the program **tcpdump** – it is your job to discover as much as possible about what Lawrence did during this session.

II. Purpose

To gain and/or enhance the following practical skills:

- Learn how to use **tcpdump** for traffic analysis
- Explore and use appropriate **tcpdump** options
- Understand the value and implications of **tcpdump**

III. Preparation

You should not attempt this task until you complete the **Credit** level task **P-Lab-04-tcpdump-C**.

IV. Methodology

After analysing Lawrence's activities and graphing the network traffic patterns generated over time for each of his online activities, you are now required to discuss its nature.

A. Discuss the nature of the graphs

For each of your four graphs (online activity session)

1) What do the graphs say about the nature of the traffic generated?
2) If there is more than one unique flow, provide insight into what is happening?
3) Is the traffic bursty/smooth? What does this mean in terms of impact on a typical home gateway?

V. Assessment

The due date for completion of practical work is **11:00pm**, exactly **six** days after your scheduled class.

Note: *The nominated submission day/time holds regardless of whether that day is a non-teaching day or public holiday*

To complete this lab you need to submit (to Doubtfire) a simple (unformatted) PDF document containing brief 1-2 sentence answers to the questions from Section IV-A.

Note: *A PDF upload is required as Doubtfire cannot currently accept uploads of image files*

A. Completion of task in Doubtfire

You will need to upload your PDF file containing your question responses to your Doubtfire portfolio before the due date

B. Tutor Discussion

In order for the submission to be marked as complete, you must discuss your work with the tutor

Unix for Telecommunications

Portfolio Task – P-Lab-09-Samba
Pass Level Task

I. Introduction

In this lab you will configure a Samba server to offer a number of shared directories on your RULE host. These directories will be privately accessible to the nominated users of those shares only.

II. Purpose

To gain and/or enhance the following practical skills:

- Deploying and configuring Samba server to offer a number of shared directories to Windows hosts
- Understand issues relating to service configuration
- Process and respond to error messages in log files
- Configure services to auto-start in Unix

III. Preparation

You can prepare for this lab by reading some of the Samba documentation available at http://www.samba.org/.

IV. Methodology

A. Samba Introduction

1) Examine the Samba web site listed above.
2) Samba has already been installed on your RULE host – how would you go about installing it under FreeBSD.
3) From reading the Samba documentation, locate:
   - The primary Samba configuration file.
   - How to start and stop Samba using the FreeBSD init scripts.
   - How to ensure Samba starts automatically at system boot.

B. Basic Samba Configuration

1) Set the workgroup name to `MSHOME`
2) Set the server string to `ruleXX.caia.swin.edu.au` (where `XX` is the number of your RULE host).
3) Set the security method to `users` and for passwords to be encrypted.
4) Ensure that the hosts allowed to access Samba shares include (at least) your Windows PC, your alternate RULE host and the RULE primary server. To simplify tasks you may wish to allow connections from any host in the Rule subnet or even from any host in the Swinburne LAN.
5) Please note that since Samba version 4.5, default values have changed regarding NTLMv1 authentication. You will now need to manually enable this in order to complete your task

C. Create Users

1) You already have the `student` and `root` accounts on your RULE host. Create a new user with the username `samba` – it is your responsibility to research how to create user accounts under FreeBSD
2) Samba maintains a password database for Samba connections that is separate to the Unix system password database. This password database is maintained using the command "`smbpasswd`". For each user account you wish to make accessible via Samba, you need to create a corresponding Samba account and password. Now create a Samba account for the user `samba`.

3) Create a second system and Samba user account with the username `autocollector` and the password `autocollector`

D. Creating Shares

Create a home share for each Samba user, the local (Unix) path to the shared directory should be "`/home/<username>`", each home directory should only be accessible to users who log on using the correct username/password combination. The authorised user should have read/write access to this share

E. Testing your Configuration

1) You should be able to connect to your Samba share from your Windows computer in the lab. Please note the following:
   - Windows will cache your password when you connect to a network share, this can be annoying when you want to connect as different users. When you wish to change the user you connect to your rule host, you need to disconnect using the command `net use \\rulexx.caia.swin.edu.au /delete`
     Note: *that Windows Explorer may cache the connection info, you may need to close Explorer, execute the above command and then restart Explorer.*
   - If the logon process does not work you should explore the Samba log files to locate why and fix the problem from there.
   - If the logon succeeds but you do not appear to have the correct access rights, again look for the Samba log entries to see what is going wrong.
2) As user `samba` you should:
   - Be able to access the `home` share.
   - Any files copied to this share (from Windows) should be placed in "`/home/samba`" on your RULE host.
3) As user `autocollector` you should:
   - Be able to access the `home` share.
   - Any files copied to this share (from Windows) should be placed in "`/home/autocollector`" on your RULE host.
4) It is your responsibility to ensure both users have full read/write access to the connected shares, that shares are mapped to the correct corresponding directories on the rule host, and that users do not have access to shares they are not supposed to have access to.

V. Assessment

The due date for completion of practical work is **11:00pm**, exactly **six** days after your scheduled class.

Note: *The nominated submission day/time holds regardless of whether that day is a non-teaching day or public holiday*

A. Self Assessment

You can self-assess your progress at any time via the marking script available at
http://ruleprimary1.caia.swin.edu.au

B. Completion of task in Doubtfire

Download the PDF output of the marking script from http://ruleprimary1.caia.swin.edu.au and submit it to Doubtfire. Your tutor will confirm completion of the lab by examing the rule marking log files on the rule server.

If you complete the task during class beforehand, you may demonstrate completion in class to your tutor.

Note: *The downloaded PDF is not evidence of successful completion of the lab, it is a document to demonstrate completion within your portfolio. Your tutor will assess the evidence via either direct confirmation via the marking script or via the log files generated when you run the marking script*

C. Tutor Discussion

In order for the submission to be marked as complete, you must discuss your work with the tutor

Unix for Telecommunications

Portfolio Task – P-Lab-09-Samba
Distinction Level Task

I. Introduction

In this lab you will add to your existing Samba configuration from **P-Lab-09-Samba-P**. You will create a third shared directory which will be publicly shared but under the control of one of the users.

II. Purpose

To gain and/or enhance the following practical skills:

- Deploying and configuring Samba server to offer a number of shared directories to Windows hosts
- Understand issues relating to service configuration
- Process and respond to error messages in log files
- Configure services to auto-start in Unix

III. Preparation

You should not attempt this task until you complete the **Pass** level task **P-Lab-09-Samba-P**.

IV. Methodology

A. Creating Shares

Without destroying/changing your existing shared directories and their access functionality:

1) Create a single public share “`public`”, the local(Unix) path to the shared directory should be “`/home/samba/pubStuff`”.
2) This share should be accessible to all users with a valid Samba account with full read/write privileges.
3) All users should have permission to edit and delete files uploaded by other users.

B. Testing your Configuration

1) As user `samba` you should:
    - Retain all existing access to the `home` share from **P-Lab-09-Samba-P**.
    - Be able to access the public share.
    - Any files copied to this share (from Windows) should be placed in “`/home/samba/pubStuff`” on your RULE host.
2) As user `autocollector` you should:
    - Retain all existing access to the `home` share from **P-Lab-09-Samba-P**.
    - Be able to access the `public` share.
    - Any files copied to this share (from Windows) should be placed in “`/home/samba/pubStuff`” on your RULE host.
3) It is your responsibility to ensure both users have full read/write access to the `public` share, that the `public` share is mapped to the correct corresponding directory on the rule host, and that access to the two pre-existing home shares is not changes from **P-Lab-09-Samba-P**.

V. Assessment

The due date for completion of practical work is **11:00pm**, exactly **six** days after your scheduled class.

Note: *The nominated submission day/time holds regardless of whether that day is a non-teaching day or public holiday*

A. Self Assessment

You can self-assess your progress at any time via the marking script available at
http://ruleprimary1.caia.swin.edu.au

B. Completion of task in Doubtfire

Download the PDF output of the marking script from http://ruleprimary1.caia.swin.edu.au and submit it to Doubtfire. Your tutor will confirm completion of the lab by examing the rule marking log files on the rule server.

If you complete the task during class beforehand, you may demonstrate completion in class to your tutor.

Note: *The downloaded PDF is not evidence of successful completion of the lab, it is a document to demonstrate completion within your portfolio. Your tutor will assess the evidence via either direct confirmation via the marking script or via the log files generated when you run the marking script*

C. Tutor Discussion

In order for the submission to be marked as complete, you must discuss your work with the tutor

Unix for Telecommunications

Portfolio Task – P-Socket-Programming
Pass Level Task

I. Introduction

In this tutorial you will develop a simple C program using the Sockets API to download and parse the contents of a web page from a remote web server.

II. Purpose

To gain and/or enhance the following practical skills:

- Basic programming skills
- Use of existing network programming tools to perform simple online tasks
- Understand the principles of communicating with remote applications

III. Preparation

You can prepare for this exercise by reviewing the Sockets API and understanding the process of establishing TCP communications with a remote host.

IV. Supporting Materials

There are two files available for download on Doubtfire for this task.

A. *wwwstat.c*

A small C program. When compiled and executed, the program will take one command line parameter consisting of a URL. The program will extract the name of the server and the name of the web page from the provided parameter. The program will call a function to perform a DNS lookup of the server and determine the IP address which is returned in a sockaddr_in structure.

The functions in this program are:

`void FillAddress(char *pcURL, struct sockaddr_in *psAddress):`

The function takes two parameters, a string containing the name of the server and a pointer a sockaddr_in structure to populate with the IP address of the server.

A DNS lookup is performed on `pcURL`, the resultant IP address is printed to screen and used to populate `psAddress`

`int main(int iArgC, char **ppcArgV):`

The main program. The function parses the command line and returns an error if there is not exactly one command line parameter.

The parameter is parsed, stripping `http://` (if it exists at the start of the string) and splitting the remained into a server name and page name. If no page name is provided (eg. `http://swin.edu.au`), then a page name of '/' is set.

The program calls `FillAddress()` to get the IP address of the server name.

B. *Makefile*

A file to allow you to execute the make command, compiling `wwwstat.c` to create the executable `wwwstat`

V. Tutorial Task

You will be extending the provided `wwwstat.c` program to perform the following tasks.

A. Connect to the web server

Use the Sockets API to establish a TCP connection to the IP address returned by the `FillAddress()` function on `WWW_PORT` (port 80)

B. Issue a HTTP request to download the specified page

Send the HTTP command to request the nominated page (`pcPage`) from the web server

C. Download the web page contents

Read from the TCP socket in a loop until the entire page contents are retrieved
Hint: Print the downloaded string to screen while it is being fetched

D. Save the downloaded page to disk

Modify the program to open a file on disk. Within the read loop save the contents of the page to the file
Hint: After the program is complete, view the saved file to confirm your program functions

VI. Bonus tasks

Finished early. Extend yourself by attempting the following tasks

A. Store entire web page in a buffer in memory

Append the downloaded text into one large memory buffer rather than save to file
Hint: Print the entire memory buffer to screen to confirm functionality

B. Parse the buffer and extract all links from the HTML source

Search the downloaded string buffer for all strings `<a href="link_text">`. When the search string is found, print the contents of `link_text` to the screen
Hint: The `strcasestr()` C function is useful for searching strings
Hint: You need to search for the initial string, then search for the end string, then print the text between the two search strings

C. Print a list of images that need to be downloaded to display the web page

Repeat the task above but this time search for the `<img src="image_tag">` string

VII. Assessment

You are expected to complete your task within the allocated tutorial time. If you are unable to complete, you will need to demonstrate partial completion by the end of the class and arrange a time with your supervisor to demonstrate task completion.

A. Assessment of Completion

Upload your completed `wwwstat.c` file to Doubtfire as evidence within your portfolio.
Your tutor will review the functionality of your code in complete downloading of a test web page. You will also be required to discuss your code functionality with the tutor to ensure your understanding.

B. Completion of task in Doubtfire

When your tutor assesses you as having both completed the understood the task, the task will be marked as complete within Doubtfire

Unix for Telecommunications

Portfolio Task – P-Lab-10-CUPS
Pass Level Task

I. Introduction

In this lab you will configure a CUPS server to offer a CUPS based network printer using your RULE host.

II. Purpose

To gain and/or enhance the following practical skills:

- Learn how to setup Printer with CUPS
- Understand how CUPS process a print job

III. Preparation

You can prepare for this lab by reading some of the CUPS documentation at http://www.cups.org.

IV. Methodology

In this lab you will create a network accessible printer with the name PDFPrinter. You will create the actual functionality of the printer in task **P-Lab-10-CUPS-D**.

A. *CUPS Introduction*

1) Examine the CUPS web site listed above
2) You should obtain a thorough understanding of how to create a printer with CUPS

B. *Installing a CUPS PDF Printer*

1) Install your CUPS PDF printer on your CUPS server. The printer name **must** be "**PDFPrinter**" and be accessible to accept print jobs from any computer on the Swinburne network (`136.186.0.0/16`)
2) Ensure that your CUPS server is configured to automatically start at system boot
3) Ensure that your CUPS server is running

V. Assessment

The due date for completion of practical work is **11:00pm**, exactly **six** days after your scheduled class.

Note: *The nominated submission day/time holds regardless of whether that day is a non-teaching day or public holiday*

A. *Self Assessment*

You can self-assess your progress at any time via the marking script available at http://ruleprimary1.caia.swin.edu.au

B. *Completion of task in Doubtfire*

Download the PDF output of the marking script from http://ruleprimary1.caia.swin.edu.au and submit it to Doubtfire. Your tutor will confirm completion of the lab by examing the rule marking log files on the rule server.

If you complete the task during class beforehand, you may demonstrate completion in class to your tutor.

Note: *The downloaded PDF is not evidence of successful completion of the lab, it is a document to demonstrate completion within your portfolio. Your tutor will assess the evidence via either direct confirmation via the marking script or via the log files generated when you run the marking script*

C. Tutor Discussion

In order for the submission to be marked as complete, you must discuss your work with the tutor

Unix for Telecommunications

Portfolio Task – P-Lab-10-CUPS
Distinction Level Task

I. Introduction

In this lab you will configure a CUPS server to offer a CUPS based network PDF printer using your RULE host.

II. Purpose

To gain and/or enhance the following practical skills:

- Learn how to setup a PDF Printer with CUPS
- Understand how CUPS process a print job
- Understand how CUPS backend and drivers work

III. Preparation

You should not attempt this task until you complete the **Pass** level task **P-Lab-10-CUPS-P**.

IV. Methodology

A. CUPS Introduction

1) Examine the CUPS web site listed above
2) You should obtain a thorough understanding of how CUPS processes a print job from:
    a) Accepting the job from a remote client
    b) Queuing the print job
    c) Building a filter chain to generate a printable format
    d) Passing the final format to the registered backend to print/output the job

B. Creating a PDF Backend

1) After creating a network printer with CUPS, you are now required to create a PDF Backend
2) Carefully read the documentation on the standard techniques for writing a CUPS backend (refer to https://www.cups.org/doc/api-filter.html and https://www.cups.org/doc/man-backend.html)
    - Standard FreeBSD executable (can be a script)
    - Required command line parameters and parameter order
    - Required output (format) from CUPS server backend query
    - Required permissions and locations to install the backend
3) You must now develop a shell script that follows the CUPS specification, the name of this script must be "`pdf`". This script should conform to the CUPS backend specifications and convert an input print job into a PDF output file.
    Hint: *Modern versions of CUPS use PDF internally as document formats so the data delivered to the backed will already be in PDF format, you basically need to only save this data to disk*
    Hint: *Do not use the pre-existing PDF drivers in the ports tree, they cannot be configured to complete this lab task as required*
4) The shell script should save all PDF jobs to disk using the following criteria:
    - PDF file saved in the directory (`/home/PDF/<username>`) where *username* is obtained from the print job
    - Any directories should be automatically created if they don't exist
    - PDF filename should be generated using the print job name and the current date/time
    - All created files and directories must be accessible by the username that generated the print job

C. Testing your CUPS PDF Printer and Backend

While the online marking script can be used to assess your lab, you should not use it for testing purposes. The printer marking script is slow and too many students using it concurrently can cause a backlog causing all assessments to fail and timeout. You should test your marking script from the command line using the command:

```
/usr/local/bin/lpr
```

You can view the man page for /usr/local/bin/lpr by executing:

```
man -a lpr
```

Note: *The first `lpr` man page shown will be for the default system `lpr` command (`/usr/bin/lpr`) while the second man page will be for the CUPS `lpr` command (`/usr/local/bin/lpr`)*

1) You should test your CUPS PDF printer from both of your allocated RULE hosts
2) Testing should be completed with PostScript files, text files and images
3) You will *not* be able to test from Windows due to you not having permissions to create a Windows printer on the lab machine. If your laptop runs Windows, you can test your printer from there

V. Assessment

The due date for completion of practical work is **11:00pm**, exactly **six** days after your scheduled class.

Note: *The nominated submission day/time holds regardless of whether that day is a non-teaching day or public holiday*

A. Self Assessment

You can self-assess your progress at any time via the marking script available at http://ruleprimary1.caia.swin.edu.au

B. Completion of task in Doubtfire

Download the PDF output of the marking script from http://ruleprimary1.caia.swin.edu.au and submit it to Doubtfire. Your tutor will confirm completion of the lab by examing the rule marking log files on the rule server.

If you complete the task during class beforehand, you may demonstrate completion in class to your tutor.

Note: *The downloaded PDF is not evidence of successful completion of the lab, it is a document to demonstrate completion within your portfolio. Your tutor will assess the evidence via either direct confirmation via the marking script or via the log files generated when you run the marking script*

C. Tutor Discussion

In order for the submission to be marked as complete, you must discuss your work with the tutor

Unix for Telecommunications

Portfolio Task – P-Project-LDAP

Pass/Credit/Distinction/High Distinction Level Task

I. Introduction

This is the major project task in your Unit and should take most of the semester to complete. In this task you will be required to design and deploy a network-based service using existing tools, and to develop and integrate a user interface to your system. The particular task required for this project revolves around deploying a networked user account authentication service with support for scripted maintenance. Details on the task can be found in this task sheet.

II. Purpose

To gain and/or enhance the following practical skills:

- Design and construct unfamiliar network services
- Synthesise knowledge gained throughout the Unit to create a systems solution
- Professional documentation of process and usage of your system

III. Scenario

The corporation in which you work has grown to the point where the number of computers, servers and corresponding computer accounts has become unmanageable. Too much time is spent by IT Support Staff to manually manage systems. In order to simplify both administration and to minimise problems such as updating passwords, they have decided to centralise the user account/password database such that all systems within the corporation use the same account/password details. The advantages include only having to remember one password and that changing a password results in an instantaneous update across all machines.

To accomplish this task, you have been assigned to deploy and configure an LDAP server to store the account/password details for the corporation and to develop a simple set of instructions to configure other machines to make use of this LDAP database. Your project is due the night of the last project presentation session (Portfolio task **C-Presentation**) in week 12.

A. Required Background Knowledge

You are required to perform research into the following topics:

- **LDAP** – How does LDAP function in general terms, and gain an understanding of the primary purposes through which it can be deployed. What are the advantages/disadvantages of LDAP systesms, particularly for authentication purposes.
- **OpenLDAP** – You will deploying OpenLDAP (http://www.openldap.org) to build your solution
- **PAM** – Pluggable Authentication Module (PAM) should be used to manage authentication and access to your Unix system. Consider how to combine this with LDAP
- **NSS** – Name Service Switch (NSS) should be used to provide user account information to your Unix systems. Consider how to combine this with LDAP
- **CLI and CGI Scripting** – Understand the options available for you to automate the process of managing your system

B. Deployment Considerations

You will be required to build, test, and deploy your system as follows:

- **Server** – Install and configure your LDAP server on your highest numbered rule host
- **Client** – Install and configure your client (remote access) on your middle numbered rule host

Note: *If you are unsure which RULE hosts you must implement your system on, ask your laboratory or tutorial supervisor.*

IV. ASSESSMENT

This section highlights the absolute requirements to achieve the nominated grade for this task

Note: *It is* ***NOT*** *required to configure an encrypted LDAP service (ldaps) for this assessment. If you are unsure as to why you don't need encryption for an authentication service, I encourage you to discuss it in the forums.*

A. Pass

In order to receive a **Pass** for this Portfolio Task, you must at a minimum successfully complete all of the following requirements.

1) LDAP Server is configured to automatically run at system boot
2) LDAP Server is correctly configured to accept requests from the Swinburne network (`136.186.0.0/16`)
3) LDAP Server is responding to authentication requests from a remote computer
4) LDAP Server is responding to account information queries from a remote computer
5) Client host is properly configured to authenticate against the LDAP server
6) Able to login to Client host using ssh and a user account only available on the LDAP Server
7) Files on the Client host owned by an LDAP user are identified as such by the filesystem (eg. `ls -l`)
8) Server Configuration Documentation – A complete list of the location and names of all configuration files you edited on the server. Also details of at least two user accounts (usernames and passwords) configured in your LDAP server
9) User Documentation – Step-by-step instructions on how `root` can add/remove user accounts from the LDAP server

B. Credit

In order to receive a **Credit** for this Portfolio Task, you must at a minimum successfully complete all of the following requirements.

1) All **Pass** requirements
2) Server host is properly configured to use itself as an LDAP authentication server
3) Able to login to Server host using ssh and a user account only available on the LDAP Server
4) Files on the Server host owned by an LDAP user are identified as such by the filesystem (eg. `ls -l`)
5) Client Configuration Documentation – Step-by-step instructions on how `root` can configure a third FreeBSD workstation to authenticate against your LDAP server. This should include:
   - List of all software to install
   - Software installation instructions and procedure
   - Configuration file changes to be made
6) Password change instructions – Step-by-step instructions how `root` can change a user's password

C. Distinction

In order to receive a **Distinction** for this Portfolio Task, you must at a minimum successfully complete all of the following requirements.

1) All **Credit** requirements
2) When the LDAP server starts/re-starts, it launches with no delays or any other warnings/errors
3) Password change instructions – Step-by-step instructions how a user can change their own password
4) A web-based or CLI solution is provided to add and delete user accounts

D. High Distinction

In order to receive a **High Distinction** for this Portfolio Task, you must at a minimum successfully complete all of the following requirements.

1) All **Distinction** requirements
2) A web-based or CLI solution to edit user account details including passwords
3) A web-based or CLI solution is provided to allow users to change their own passwords
4) Use your imagination for other features you may want to add to your system. One example might be scripting to automate installation of a new FreeBSD client host
   Note: *Completing the minimal set of requirements will only guarantee you a minimal **High Distinction** result*

V. Self Assessment

Below are the steps we will take when assessing your system. This is a complete set of tests that will confirm functionality of all level tasks, it is up to you to determine whether you meet your target result.

A. LDAP Server

From your rule host:

- Check that `slapd` was automatically started
- Check that startup occured without any delay or any other warnings/errors
- Examine your configuration files to ensure you have correctly secured your database
- Confirm that **all** LDAP user accounts do **NOT** exist in the passwd file
- Run `getent` to retrieve user information

From a desktop computer

- Probe your LDAP server to ensure that queries work from another Swinburne machine
- Perform an LDAP query to extract user account information
- **ssh** into your server using an LDAP account with the provided password
- Create a file owned by an LDAP user and check ownership is correctly reported via `ls -l`

B. LDAP Client

- Check that `slapd` is **NOT** running
- Examine your configuration files to ensure client is configured to use your LDAP server host
- Confirm that **all** LDAP user accounts do **NOT** exist in the passwd file
- Run `getent` to retrieve user information

From a desktop computer

- **ssh** into your client using an LDAP account with the provided password
- Create a file owned by an LDAP user and check ownership is correctly reported via `ls -l`

C. Other Tasks

Other tasks we will perform to verify functionality include:

- An LDAP user will be created, tested, and deleted
- A third machine will be built to authenticate against your LDAP server
- The password for one user will be modified
- An attempt to change the password of the other user should fail

D. Implemented code

The code will be examined to see how it works and whether it has been properly structured and commented

E. Documentation

The documentation will be checked to ensure that it is complete (*all required documentation has been submitted*) and is correct (*the nominated instructions work as provided*)

VI. Assessment

For task **C-Project-LDAP**, you will be graded to a **Pass**, **Credit**, **Distinction** or **High Distinction** level as per the requirements listed above.

A. High Distinction Assessment

A Project graded at a **High Distinction** level will also be given a score based on the overall quality of the solution and any extra features you have decided to implement. Students will be given a score of **HD1** (meets minimum HD requirements), **HD2** (*Excellent work, cool new features, well implemented*), or **HD3** (*Did you hire a professional to build this*!!!). If a student is eligible for a **HD** result on their Portfolio, this result will be used as partial input to determine the student final score between **80HD** and **100HD**.

VII. Submission

The due date for completion of the project is **11:00pm** on the night of the **final** project presentation session in **Week 12**.

You must upload your documentation and select the Ready for Feedback option in your Doubtfire portfolio by the due date. Once this is done, your RULE hosts will be disabled until marking is complete.

A. Completion of task in Doubtfire

Your Doubtfire result will be updated once your RULE host has been assessed.

Unix for Telecommunications

Portfolio Task – P-Project-Bind

Pass/Credit/Distinction/High Distinction Level Task

I. Introduction

This is the major project task in your Unit and should take most of the semester to complete. In this task you will be required to design and deploy a network-based service using existing tools, and to develop and integrate a user interface to your system. The particular task required for this project revolves around deploying a prototype dynamic DNS service with support for multiple zones and users. Details on the task can be found in this task sheet.

II. Purpose

To gain and/or enhance the following practical skills:

- Design and construct unfamiliar network services
- Synthesise knowledge gained throughout the Unit to create a systems solution
- Professional documentation of process and usage of your system

III. Scenario

The corporation in which you work runs and maintains a DNS server which it uses to offer DNS services to customers. Until recently, requested changes to the DNS database from your customers has been minimal and handled via a simple request system:

- The customer asks for a change to the database
- You stop the DNS server
- The zone database files are updated
- The DNS server is restarted

Update requests have recently increased to the point where this system has become unmanageable. Your boss has asked to you to investigate a system that allows the database to be updated dynamically:

- The DNS server does not need to be stopped
- Updates can be performed via a scripted and/or web-page approach
- Access can be given to customers to update their own entries

To accomplish this task, you have been assigned to deploy and configure a BIND server with dynamic update capability, along with a series of scripts and/or CGI based web pages to update the database contents. Your assignment is due the night of the last project presentation session (Portfolio task **C-Presentation**) in week 12.

A. Required Background Knowledge

You are required to perform research into the following topics:

- **DNS and BIND** – How does DNS function in general terms, and how is BIND configured to participate in the DNS distributed database.
- **Dynamic DNS** – How does Dynamic DNS function and how does BIND currently support this functionality
- **Multi-User Support** – Consider how to use the features of both BIND Dynamic Updates, general Unix security, and other mechanisms to support securing DNS zones such that updates can only be performed by the authorised user/customer
- **CLI and CGI Scripting** – Understand the options available for you to automate the process of updating the database

B. Deployment Considerations

You will be required to build, test, and deploy your prototype solution on your *(highest or middle)-numbered* RULE host along with Apache and any scripts you develop.

Note: *If you are unsure which RULE host you must implement your system on, ask your laboratory or tutorial supervisor.*

The organisation also owns the IP address subnet **`136.186.230.0/24`**. If a host is added or removed from the DNS and has an IP address in this subnet, then you should also add/remove the corresponding entry from the reverse lookup zone as well.

IV. Assessment

This section highlights the absolute requirements to achieve the nominated grade for this task

A. Pass

In order to receive a **Pass** for this Portfolio Task, you must at a minimum successfully complete all of the following requirements.

1) Your RULE host is configured to use itself as a DNS server
2) BIND Server is correctly configured to accept dynamic update requests
3) At least two (custom) DNS Zones and the reverse lookup zone are configured to support dynamic updates. It is required that you support a reverse lookup zone for the subnet `136.186.230.0/24`
4) All unknown requests/lookups for names it is not responsible/authoritative for are forwarded to the Swinburne DNS server
5) Either a Web or CLI based tool/script to allow adding and removing entries to the database
6) Adding an entry with an IP address in the range will result in the reverse update zone being updated
7) Developed Software Documentation – A complete list of the location and names of all scripts you have developed
8) User Documentation – Step-by-step instructions on how to add/update/remove DNS entries using all tools (Web and/or CLI) you have developed

B. Credit

In order to receive a **Credit** for this Portfolio Task, you must at a minimum successfully complete all of the following requirements.

1) All **Pass** requirements
2) Your DNS server is configured to accept public DNS queries (from hosts external to your RULE host)
3) Public queries for non-authoritative zones are forwarded to the Swinburne DNS server and then responded to
4) When removing a host from your zone, it will automatically determine the IP address and if it is in the **`136.186.230.0/24`** subnet, remove it from the corresponding reverse lookup zone

C. Distinction

In order to receive a **Distinction** for this Portfolio Task, you must at a minimum successfully complete all of the following requirements.

1) All **Credit** requirements
2) Updating host details correctly functions for all cases of
   - Updating a host with an address `136.186.230.*` to a new address of `136.186.230.*`
   - Updating a host with an external address to a new external address
   - Updating a host with an external address to a new address of `136.186.230.*`
   - Updating a host with an address `136.186.230.*` to a new external address
3) Both a web-based and CLI solution to automating updates has been provided

D. High Distinction

In order to receive a **High Distinction** for this Portfolio Task, you must at a minimum successfully complete all of the following requirements.

1) All **Distinction** requirements
2) Support for multiple users where each user is allowed to update their own zone
3) All users can update the reverse zone
4) Multi-User Documentation – Multi-user solution documentation including:
   - User account details including all usernames and passwords
   - Which zones each user is allowed to update
5) Use your imagination for other features you may want to add to your system. One example might be documentation/scripting to create a new account/zone for a new customer
   Note: *Completing the minimal set of requirements will only guarantee you a minimal* ***High Distinction*** *result*

V. Self Assessment

Below are the steps we will take (not necessarily using the exact zones and host names listed here). This is a complete set of tests that will confirm functionality of all level tasks, it is up to you to determine whether you meet your target result.

A. BIND Server

From your rule host:

- Check that `named` was automatically started
- Check `named.conf` that at least two forward zones are correctly configured for dynamic updates and the `136.186.230.*` reverse zone is correctly configured
- Check no errors on starting `named` and that the dynamic zones are functioning
- `nslookup www.google.com` (*or some external address*)
- `nslookup foo.zone1` (*or any host that exists in one of the zones in your DNS server*)
- `nslookup 136.186.230.5` (*or any IP address that is in your reverse zone*)

From my desktop computer (**Credit** level task)

- Configure to use your RULE host as a DNS server
- Repeat three nslookup tests from above

B. Automated tools

Assuming you have two dynamic zones `.zone1` and `.zone2`

- Add `host1.zone1` → `1.2.3.4`
  `nslookup host1.zone1` – Should return `1.2.3.4`
  `nslookup 1.2.3.4` – Should be forwarded to the Swinburne DNS server
- Add `host2.zone2` → `136.186.230.10`
  `nslookup host2.zone2` – Should return `136.186.230.10`
  `nslookup 136.186.230.10` – Should return `host2.zone2`
- Update `host1.zone1` → `5.6.7.8`
  `nslookup host1.zone1` – Should return `5.6.7.8`
  `nslookup 5.6.7.8` – Should be forwarded to the Swinburne DNS server
- Update `host1.zone1` → `136.186.230.11`
  `nslookup host1.zone1` – Should return `136.186.230.11`
  `nslookup 136.186.230.11` – Should return `host1.zone1`
- Update `host2.zone2` → `136.186.230.12`
  `nslookup host2.zone2` – Should return `136.186.230.12`

`nslookup 136.186.230.10` – Should fail with no answer
`nslookup 136.186.230.12` – Should return `host2.zone2`

- Update `host2.zone2` → `9.9.9.9`
  `nslookup host2.zone2` – Should return `9.9.9.9`
  `nslookup 136.186.230.12` – Should fail with no answer
  `nslookup 9.9.9.9` – Should be forwarded to the Swinburne DNS server
- Delete `host1.zone1`
  `nslookup host1.zone1` – Should fail with no answer
  `nslookup 136.186.230.11` – Should fail with no answer
- Delete `host2.zone2`
  `nslookup host2.zone2` – Should fail with no answer
- Tests will be performed with both CLI and CGI interfaces if they exist

C. Implemented code

The code will be examined to see how it works and whether it has been properly structured and commented

D. Documentation

The documentation will be checked to ensure that it is complete (*all required documentation has been submitted*) and is correct (*the nominated instructions work as provided*)

VI. Assessment

For task **C-Project-Bind**, you will be graded to a **Pass**, **Credit**, **Distinction** or **High Distinction** level as per the requirements listed above.

A. High Distinction Assessment

A Project graded at a **High Distinction** level will also be given a score based on the overall quality of the solution and any extra features you have decided to implement. Students will be given a score of **HD1** (meets minimum HD requirements), **HD2** (*Excellent work, cool new features, well implemented*), or **HD3** (*Did you hire a professional to build this*!!!). If a student is eligible for a **HD** result on their Portfolio, this result will be used as partial input to determine the student final score between **80HD** and **100HD**.

VII. Submission

The due date for completion of the project is **11:00pm** on the night of the **final** project presentation session in **Week 12**.

You must upload your documentation and select the Ready for Feedback option in your Doubtfire portfolio by the due date. Once this is done, your RULE host will be disabled until marking is complete.

A. Completion of task in Doubtfire

Your Doubtfire result will be updated once your RULE host has been assessed.