Experimental Intranet Services over Swinburne's 802.11b Wireless Infrastructure.

Paul A. van den Bergen, Grenville J. Armitage
Centre for Advanced Internet Architectures. Technical Report 030730A
Swinburne University of Technology
Melbourne, Australia
pvandenbergen@swin.edu.au, garmitage@swin.edu.au

Abstract- **This paper falls under the MAGIC project which deals with mobile wireless networks. As part of the MAGIC project, we wished to advertise our presence and demonstrate the sorts of services wireless technology can provide. Swinburne University has as part of its IT infrastructure a wireless network covering the Hawthorn Campus. It is worth leveraging this infrastructure to further the research goals of the group.**

To this end, an EPIA mini-ITX PC was configured using FreeBSD to provide local Intranet services to the Swinburne Wireless Infrastructure (Swinnet). Swinnet consists of a number of Access Points distributed around campus. The 802.11b traffic is bridged between Access Points through the Swinburne ethernet backbone on a specific VLAN. To enable the desktop PC to access this traffic, that VLAN traffic was trunked to the particular wall socket to which the PC is attached. Native ethernet packets are therefore presented to (or from) the desktop box so as it appears to reside on the 802.11b network ethernet segment.

Keywords- 802.11b, CAIA, FreeBSD, MAGIC, VIA EPIA mini-ITX, Wireless Network.

I. Introduction.

This technical report outlines an experiment conducted at CAIA to evaluate providing localised IP services using the Wireless Infrastructure at Swinburne. This falls under the auspices of the MAGIC project (Mobile Applications and Global Internet Communications)[1].

The wireless infrastructure at Swinburne consists of a number of wireless Access Points connected to the Swinburne ITS switched backbone. Authentication is provided by IPSec (Cisco's windows-based VPN solution) and WEP is not implemented. The 802.11b traffic is bridged between APs over the 802.3 ethernet using the university backbone network on a single vlan (802.1Q tagged frames).

Since the traffic intended for the insecure wireless network is available to the entire Swinburne backbone, any box appropriately configured and connected to that specific internal vlan appears available to the wireless ethernet segment. One can therefore leverage the wireless infrastructure to provide additional IP-based services, in parallel with regular access to Swinburne's network. The intent was to provide a few intranet services as a demonstration of the possibilities and promote the center's research capabilities.

A single PC was configured with 4 virtual hosts, each with a different (aliased) IP address. Each virtual host (referred to as jail hosts for the rest of this paper) contained one of; a dns server for our own unofficial top level domain (TLD), wireless.caia; an ftp server providing access to FreeBSD install disks and CDROM iso images; a website (a simple copy of the official CAIA MAGIC homepage) and a game server running a halflife game environment. The traffic for these 4 sites was trunked over Swinburne's network and made available on the wireless network (i.e. the traffic was bridged such that it was on the same ethernet segment).

The dns server runs **`bind`** and defines our own private top level domain, ".caia", of which "wireless.caia" is the only existing network.

The webserver runs apache 1.3 and a simplified version of the official project webpage[1].

The ftp server allows anonymous ftp access to FreeBSD 4.x iso images and the full FreeBSD 4.x file system.

The gameserver is running **`hlds`**, the half life dedicated server for Linux.

The operating system used was FreeBSD[2].A number of FreeBSD utilities are used to enable this project, including **`jail`**, **`vlan`**, **`ipfw`** and **`dummynet`**.

- **`jail`**

  A FreeBSD kernel utility which imprisons a process and all it's descendants.

- **`ipfw`**

  A kernel implemented IP packet firewall.

- **`dummynet`**

  An add-on to **`ipfw`** that allows traffic specified by **`ipfw`** rules to be piped to a ruleset that specifies traffic shaping parameters.

- **`vlan`**

  IEEE 802.1Q VLAN network interface.

`jail` allows virtualisation of the kernel functionality. This is often used as a security feature, where a jailed virtual host is provided within which to run network services such as ftp, web servers, etc. Due to restrictions in the kernel function calls imposed on the virtualise host by the kernel, compromising the root login on the

jailed host does not immediately compromise the entire system.

Within CAIA, **jail** has been used to provide multiple instances of FreeBSD to students, allowing them to get hands on experience of system administration[3]. Root (system administration) access to multiple instances of FreeBSD can be provided on relatively few machines and “destroying” a given host is no worse (with the exception of bruised egos) than getting a host administrator to reset (tear down and reinstall) a particular host.

In the current project **jail** is used to implement virtual hosts that themselves supported a range of additional networked services.

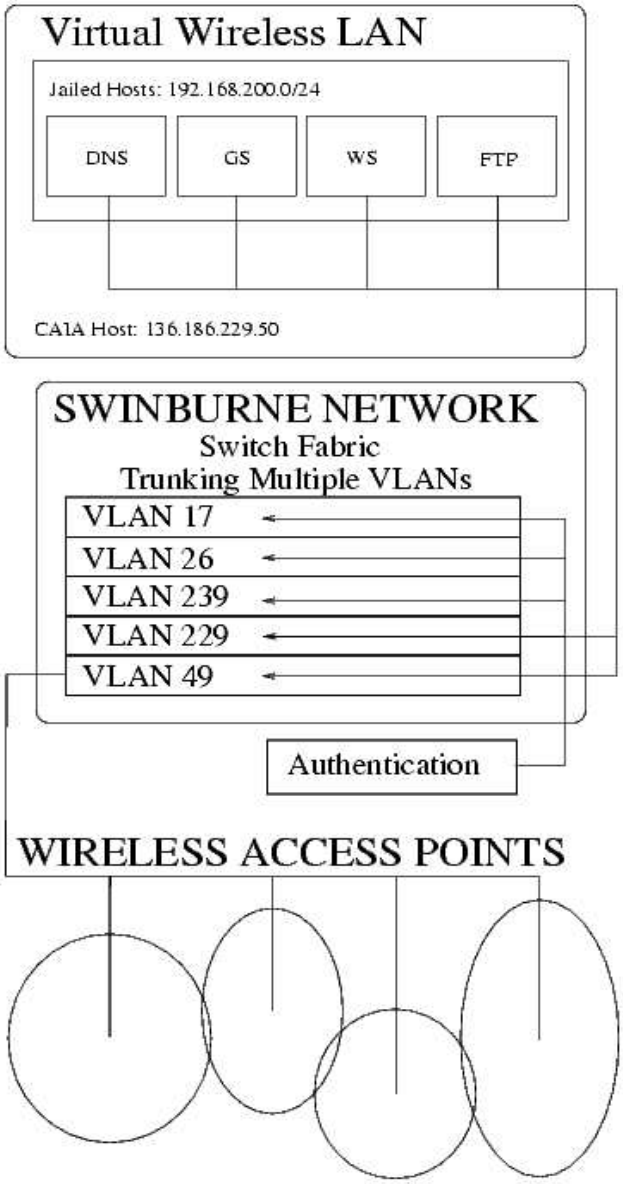

Figure 1: Conceptual Trunking Layout for CAIAs attatchment to Swinburne's Wireless Service

ipfw and **dummynet** are used to allow traffic shaping (bandwidth limiting) of the traffic to and from the ftp and gameserver hosts.

vlan enables 802.1Q virtual interfaces over a regular physical ethernet interface, in this case allowing us to access Swinburne's wireless and wired networks separately from the same host.

II. Quick Guide for the Lazy and Impatient.

To connect to the private network (our “fake” top level domain, “.caia”) and use the available services, one needs the following.

An 802.11b wireless device within range of the Swinburne's WiFi network.

802.11b (WiFi) settings; SSID = swinnet, channel 6

DNS = 192.168.200.254

default gateway = 192.168.200.33

IP address = 192.168.200.100-199/24

Ping your chosen address to check for duplicates as DHCP has not been implemented for this experiment.

Use a web browser to look at webserver.wireless.caia, ftp can be accessed on ftp.wireless.caia and the halflife gameserver can be accessed on gs.wireless.caia. Note that this is our private domain structure and is only accessible via Swinburne's wireless network and is not accessible via the public internet.

III. Hardware.

The system consist of;

- VIA EPIA 533MHz mini-ITX motherboard[4] fitted with one 256MB SDRAM DIMM.
- 80 GB HDD Seagate Barracuda 7200.7 ST380011A [5].
- Intel-based PCI LAN card.

A. Hardware Specifics.

The VIA EPIA motherboard is an ATA PSU compliant motherboard based on the Embedded VIA Eden ESP 5000 processor. It uses the VIA VT8231 South bridge chip and the VIA Apollo PLE133 North bridge chip. It has a number of the standard all-in-one on board integrated peripherals: Trident Blade 3D graphics, 10/100 ethernet (VT6103), sound (VT1612A AC97 Codec), dual USB, serial, parallel, PS2 mouse and keyboard, a single PCI slot, 2 IDE Ports (ATA100) and 2 PC66/100/133 SDRAM DIMM slots (up to 1GB). In addition, it has S-video and RCA or S/P DIF TV/video out ports(VT1621, supporting 640x480, 800x600, PAL, NTSC). Its low power consumption (47W max, 60W peak) means it requires no fan for cooling, making it very quiet.

The Seagate Barracuda 7200.7 80GB HDD (ST380011A) is use on the system. It operates at 7200 rpm and is capable of peak data transfer rates of 100MBps using UDMA mode 5.

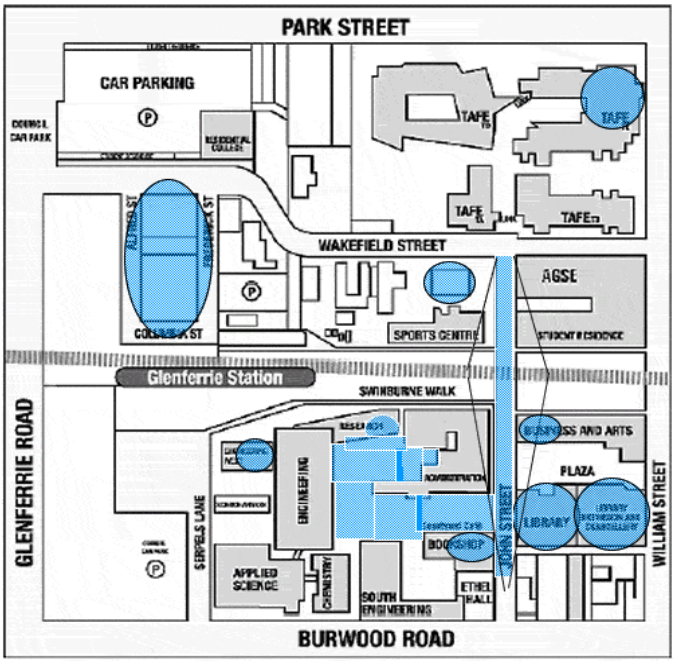

Figure 2: Swinburne Hawthorn Campus Wireless Coverage

The file structure of the HDD is as follows. The HDD is formatted into 3 slices. The first slice is 5GB at the start of the disk. This is currently empty and is to allow for the installation of other operating systems if required in future. The second slice is 10 GB and contains the FreeBSD primary host installation. The third slice uses the remainer of the disk and is broken into 10GB partitions to accommodate the jails.

The LAN adapter is manufactured by Kelex (K668 MP721502-006) based on the Intel i82559 chip set[6] and supported using the fxp driver[7][8]. This was used in preference to the on board VIA chip set because the VIA ethernet port (vr0) has problems with vlan tagged packets. 802.1Q vlan tagged packets require an additional 4 bytes, which the VIA ethernet card does not seem to be able to handle correctly.

IV. Trunking Wireless Services

A. *Swinburne's Wireless Network Layout*

Details on Swinburne's wireless roll out can be found on the Swinburne ITS web page[9]. Figure 2 shows the wireless coverage of the various APs around the campus[10]. This is expected to expand to cover the entire University in the future.

B. *CAIA Connection Configuration.*

Figure 1 shows the connection of the PC to the Swinburne backbone network. A specific wall socket was configured so that packets for the wireless network (VLAN 49 in Figure 1) were delivered directly to and from that port as native 802.3 ethernet packets. Packets for the wired Swinburne network are forwarded to the same port, but as 802.1Q frames tagged for VLAN 229 (Figure 1).

C. *Security*

No packet forwarding is allowed between the Swinburne wireless network and the Swinburne internal wired network.

Security is achieved by the following methods

- Network routing on the host is not activated
- Jailed hosts are used to isolate the on line services
- Access to the primary host is only via **`ssh`** or physically via the console.

Since packet forwarding by the primary host host is deactivated, routing cannot occur. The **`ipfw`** firewall is used in conjunction with **`dummynet`**, to provide network traffic shaping, impose a bandwidth limit on specific hosted services.

From within Swinburne, access to the primary host is via **`ssh`** only and access to the jailed hosts is via **`ssh`** from the primary host, a two step process. Both activities require access to a non-root account on the respective host machine. Access from the Swinburne wireless network is only permitted to the services running on the jailed hosts. Breaking the security of the jail, including gaining root access on the jailed host, would not result in compromising the accounts on the primary host.

The security of this local private network is reasonable and does not represent a back door into Swinburne's wired network.

D. *Configuration - Jails.*

A number of jailed hosts were set up using an in house tool set, known as the jail host toolkit (**`jht`**)[3]. The **`jht`** assists in the configuration and administration of jailed hosts.

Below are listed the system configuration changes performed by **`jht`** to set up the jailed hosts.

- For each virtual host to be mounted on the primary host file system, a single partition is required.
- **`jht`** will copy the primary host configuration to the empty partition and mount it at the appropriate point in the primary hosts file system, specifically **`/home/<jail IP address>/`**
- The **`jht`** can use the host name to mount the associated IP address attachment point. In this case access to a DNS or the correct entires in **`/etc/hosts`** is required.
- The **`jht`** configures the specified interface with the appropriate IP address alias.

`ssh` is used to connect to the virtual host. As **`ssh`** will generally not allow direct root logins, the jailed host must have a non-root user available, with membership of the wheel group to allow the user to **`su`** to root.

The IP address range selected for the publicly available jailed Internet services on the wireless ethernet segment is the private address space 192.168.200.0/24.

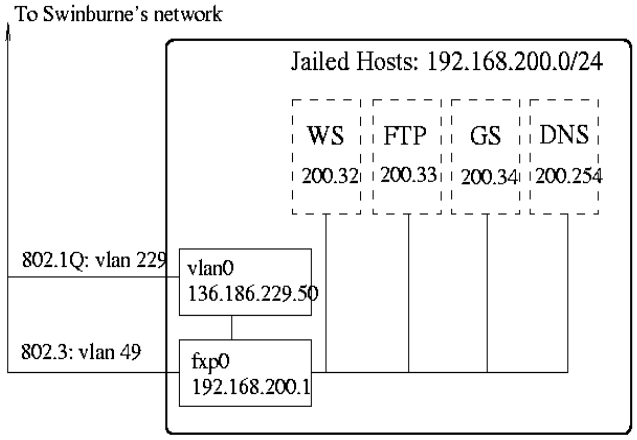

Figure 3: Jailed Host Configuration.

E. Configuration - Primary Host.

The primary host has a separate identity in the university's IP address space. This is implemented as a vlan on an appropriate interface (VLAN 229 in Figure 1). The security considerations of this arrangement have already been discussed. The primary host can only be accessed by direct console logon to the host machine from the university side or by ssh via the CAIA network. The wireless side of the connections can only see the jailed hosts. To totally isolate the primary host, one could disable **ssh** access to it, hence allowing physical console access as the only method of access.

For ease of implementation, a custom kernel was created. This enabled **ipfw, dummynet** and **vlan** functionality to be installed in one process. **ipfw, dummynet** and **vlan** can all be implemented at run time by a number of methods, e.g. using key words in **/etc/rc.conf**, loading kernel modules using **kldload**, running boot-time scripts from **/usr/local/etc/rc.d/** or custom compiling a new kernel. Most of the associated settings and keywords used in the kernel or boot time configurations files have equivalents at run time, with a few exceptions with respect to **ipfw6** – the IPv6 equivalent of **ipfw**. Initially, ipv6 will not be needed, but future use is planned so the selected method for installing **ipfw, dummynet** and **vlan** were compiled into a new kernel. The details of customising the kernel are beyond the scope of this paper but several good resources are available[2].

The primary host needs to be configured so that the jailed services will come up on boot. This is achieved as part of the **jht** configuration, with the exception that **jht** uses the dns lookup to determine the names and IP addresses to identify the jails. Since it does not know the dns entries for the jails before the dns server is booted, the entry for the dns server at least must exist in the primary hosts **/etc/hosts** file, which by default is checked before the dns server daemon, named.

This means that boot order of the jails is important – the dns server needs to boot completely first before the other jails can boot. The **jht** can be configured so that the jail boot order is specified. If a dns server is not required, the dns entries can all be specified in **/etc/hosts**. The **/etc/hosts** file contains the following entries as a minimum requirement to allow the boot scripts to find the dns server address.

```
127.0.0.1
   localhost.wireless.caia
   localhost
192.168.200.254
   dns.wireless.caia
   dns
136.186.229.16
   caia.swin.edu.au
```

F. Ethernet Interface Configuration.

The **jht** will correctly configure the aliases on the ethernet interface. After installing the jails, **ifconfig** on the host box has the following settings.

```
fxp0:
   flags=8843<UP,BROADCAST,RUNNING,
         SIMPLEX,MULTICAST> mtu 1500
   inet 192.168.200.254
         netmask 0xffffffff
         broadcast 192.168.200.254
   inet 192.168.200.32
         netmask 0xffffffff
         broadcast 192.168.200.32
   inet 192.168.200.34
         netmask 0xffffffff
         broadcast 192.168.200.34
   inet 192.168.200.33
         netmask 0xffffffff
         broadcast 192.168.200.33
   ether 00:02:b3:0a:b0:1f
   media: Ethernet autoselect
         (100baseTX <full-duplex>)
   status: active
```

To allow access to the Swinburne network services a vlan interface is created.

```
vlan0: flags=8843<UP,BROADCAST,RUNNING,
         SIMPLEX,MULTICAST> mtu 1500
   inet 136.186.229.50
         netmask 0xffffff00
         broadcast 136.186.229.255
   ether 00:02:b3:0a:b0:1f
   vlan: 4
   parent interface: fxp0
```

This is set using a line inserted into **/etc/rc.conf**.

```
ifconfig_vlan0="create
         inet 136.186.229.50
         netmask 255.255.255.0
         vlan 4 vlandev fxp0"
```

This is shown schematically in Figure 3. Each jailed host, representing a separate virtual host, has it's own IP address, which is a single alias of the primary physical ethernet interface. The ethernet connection is to a local Swinburne switch, which is configured to provide vlan 49 traffic as 802.3 ethernet frames and all other available traffic as 802.1Q tagged vlan frames. Access to VLAN 229 (see Figure 1) is via the virtual device **vlan0**.

G. Services Provided

Four services were provided, DNS (dns.wireless.caia, 192.168.200.254), an apache server (webserver.wireless.caia, 192.168.200.32) , an ftp server (ftp.wireless.caia, 192.168.200.33) and a halflife gameserver (gameserver.wireless.caia, 192.168.200.34).

DNS

The dns server runs the BIND daemon name service, **/usr/sbin/named**. The DNS configuration file (**/etc/namedb/s/db.wireless.caia**) has the following entries.

```
@ 864000 IN SOA dns.wireless.caia.
   root.wireless.caia.
   (200103021 10800 3600 604800 86400 )
   @  IN A  192.168.200.254
   @  IN NS dns.wireless.caia.
   wireless.caia.
         IN NS        dns.wireless.caia
   localhost.wireless.caia.
         IN A         127.0.0.1
   host.wireless.caia.
         IN A         192.168.200.31
   webserver.wireless.caia.
         IN A         192.168.200.32
   ftp.wireless.caia.
         IN A         192.168.200.33
   gameserver.wireless.caia.
         IN A         192.168.200.34
   dns.wireless.caia.
         IN A         192.168.200.254
   ws.wireless.caia.
         IN CNAME webserver.wireless.caia.
   gs.wireless.caia.
         IN CNAME gameserver.wireless.caia.
```

and the reverse DNS configuration file (**/etc/namedb/s/db.200.168.192.in-addr.arpa**) has the following settings.

```
$TTL 604800
200.168.192.in-addr.arpa. IN SOA
   dns.wireless.caia
   root.wireless.caia.
   (200103021 10800 3600 604800 86400)
;Name servers
200.168.192.in-addr.arpa.
         IN NS dns.wireless.caia.
;Reverse name lookups
31.200.168.192.in-addr.arpa.
         IN PTR  host.wireless.caia.
   32.200.168.192.in-addr.arpa.
         IN PTR webserver.wireless.caia.
33.200.168.192.in-addr.arpa.
         IN PTR ftp.wireless.caia.
34.200.168.192.in-addr.arpa.
         IN PTR
   gameserver.wireless.caia.
254.200.168.192.in-addr.arpa.
         IN PTR dns.wireless.caia.
```

We have thus created a private domain name web space based on the private address range 192.168.200.0 – 254. and the top level domain wireless.caia.

FTP

The ftp server runs **ftpd**. Available for download are the FreeBSD CD iso images and the contents of those iso images.

The ftp server is configured as an anonymous server for download and does not accept uploads[11]. It is bandwidth limited to 1 Mbps via the host interface using **dummynet**.

Web Server

Apache 1.3[12] was implemented on the virtual host webserver.wireless.caia (192.168.200.32). Since this is essentially for demonstration purposes, it is serving as a mirror for CAIA MAGIC home page.

Game Server

The game server host, gs.wireless.caia (192.168.200.34), is running a HalfLife[13] server (**hlds** - Half-Life Dedicated Server[14]). This is a Linux[15] binary running on a FreeBSD virtual host. Some specific configuration issues, beyond the current discussion, is need for this binary to run. This server is bandwidth limited to 500 kbps.

Firewall Configuration – Primary Host

A firewall (**ipfw**) is utilised on the primary host to minimise security problems.

A number of articles dealing with configuring a firewall are available on line[16]. The following set of ipfw rules were developed on the basis of that guide.

To enable the firewall, the following lines are added to **/etc/rc.conf**.

```
firewall_enable="YES"
firewall_type="/etc/ipfw.rules"
```

The file **/etc/ipfw.rules** is a user-created file which contains the firewall rules. The contents of ,**/etc/ipfw.rules** is listed below, with comments stripped out for berevity.

```
-f flush
pipe 1 config bw 1000Kbit/s
pipe 2 config bw 1000Kbit/s
add 100 allow all
   from any to any via lo0
add 1000 pipe 1 tcp
   from 192.168.200.33
   to 192.168.200.0/24 established
add 1100 allow tcp
   from 192.168.200.0/24
   to 192.168.200.0/24 established
add 2000 deny tcp
   from 192.168.200.0/24
   to 192.168.200.1 22 setup
add 2100 allow tcp
   from 192.168.200.0/24
   to 192.168.200.0/24 22 setup
add 2200 pipe 1 tcp
   from 192.168.200.0/24
   to 192.168.200.33 21 setup
add 2300 allow tcp
   from 192.168.200.0/24
   to 192.168.200.32 80 setup
add 3000 deny icmp
   from 192.168.200.0/24
   to 192.168.200.1
add 3100 allow icmp
   from 192.168.200.0/24
   to 192.168.200.0/24
add 4000 deny udp
   from not 136.186.229.0/24
   to not 136.186.229.50
   6000-6063,111,514
```

```
add 4100 deny udp
   from not 136.186.229.0/24
   6000-6063,111,514
   to not 136.186.229.50
add 4200 pipe 2 udp
   from 192.168.200.0/24
   to 192.168.200.34
add 4300 pipe 2 udp
   from 192.168.200.0/24
   to 192.168.200.34
add 4400 allow udp
   from 192.168.200.0/24
   to 192.168.200.254 53
add 4500 allow udp
   from 192.168.200.254 53
   to 192.168.200.0/24
add 10000 allow tcp
   from 136.186.229.0/24
   to 136.186.229.0/24 established
add 11000 allow tcp
   from 136.186.229.0/24
   to 136.186.229.0/24 22 setup
add 12000 allow udp
   from 136.186.229.0/24
   to 136.186.229.0/24
add 13000 allow tcp
   from 136.186.229.0/24
   to 136.186.229.0/24 2049
add 13100 allow tcp
   from 136.186.229.0/24 2049
   to 136.186.229.0/24
add 14000 allow icmp
   from 136.186.229.0/24
   to 136.186.229.0/24
```

Initialisation

The default behaviour is to deny-all, so the final ipfw rule, 65535, is automatically set to deny ip from any to any. Any traffic not matching the preceeding rules (as specified above) will be discarded. The rules are initially flushed to clear any pre-existing settings.

Two pipes are established to allow traffic shaping. The traffic transferred to these pipes is bandwidth limited to 1Mbps.

Rule 100 allows local loopback operations. In general, rules from 1,000 up to 9,999 deal with connections from the wireless network – wireless.caia or 192.168.200.0/24 - and rules from 10,000 to 19,999 deal with the wired network – caia.swin.edu.au or 136.186.229.0/24.

Wireless Network – 192.168.200.0/24 (wireless.caia)

Rule 1,000 and 1,100 allows established tcp connections. Rule 1,000 separates traffic (e.g. using **`dummynet`**) originating from the ftp server (192.168.200.33) into pipe 1 so that a 1Mbps bandwidth limit applies on the ftp server traffic. Rule 1,100 enables all other established tcp sessions on the wireless network.

Rule 2,000 prevents ssh access (port 22) to the host machine from the wireless network. Rules 2,100, 2,200 and 2,300 allow specific tcp connections to be established. Any **`ssh`** session (port 22) on the wireless network (excluded rule 2,000 matches), ftp (port 21) to ftp.wireless.caia (via pipe 1) and http (port 80) to webserver.wireless.caia, respectively, are permitted.

Rules 3,000 and 3,100 allow icmp traffic to permit ping on the wireless network, with the exception of the primary host.

Known potentially vulnerable udp ports are; udp port 111 is the sunrpc portmapper; udp port 514 is syslog and udp port 6000 – 60063 is the x11 server. These should be blocked (rules 4,000 and 4,100) before allowing udp traffic related to the game server (rules 4,200 and 4,300). The only exception to this is udp traffic on the wired network (136.186.229.0/24) destined for the primary host, wlan1.caia.swin.edu.au (136.186.229.50).

The gameserver traffic goes through pipe 2 to bandwidth limit the traffic to 1Mbps.

Rules 4,400 and 4,500 allow dns (port 53) traffic to reach and leave the dns server.

Wired Network – 136.186.229.0/24 (caia.swin.edu.au)

Similar to the wireless network, established tcp sessions are allowed, by rule 10,000, on the wired network. **`ssh`** setup is allowed by rule 11,000.

There are several services on the wired network that use udp. Rule 12000 allows udp traffic for all of caia.swin.edu.au.

Nfs services (rule 13,000 and 13,100) and ping (rule 14,000) are required as part of the configuration of the jails on the wired network, during both setup and booting.

V. Conclusions

The purpose of this exercise was to demonstrate the capacity of FreeBSD-based PCs to leverage the Swinburne wireless infrastructure and extend the types of services available over Wireless. The exercise itself was relatively straight forward and easy to accomplish. Security issues were dealt with using physical disconnection and **`ipfw`** firewall.

The experimental IP services can be accessed from anywhere Swinburne's Wireless Network on the Hawthorn Campus can be detected (with sufficient strength to connect). This does not allow access to Swinburne's internal network and one cannot access these services from Swinburne's internal network.

The method of accessing the services are as follows.

1) Set your 802.11b wireless settings to;
   1. SSID = “swinnet” (without quotes)
   2. Channel = 6

      (alternatively use a wireless discovery program, such as kismet[17] or netstumbler[18] to determine the correct settings)
2) Set your network settings to;
   1. Proxy – Direct connection to network (no proxy)
   2. DNS = 192.168.200.254
   3. Default Gateway = 192.168.200.33

4. IP Address = 192.168.200.100-199 (be aware that since we are not running dhcp, there is a chance that duplicate IP addresses may occur.)

With these settings one should be able to see the webpage at http://webserver.wireless.caia, browse the ftp files at ftp://ftp.wireless.caia and connect to the halflife game server at gameserver.wireless.caia.

Enjoy

VI. References.

[1] "Mobile Applications and Global Internet Communications", (as at 28/7/20003), http://www.caia.swin.edu.au/magic/

[2] "The FreeBSD Project", (as at 28/7/20003), http://www.freebsd.org/

[3] G.J. Armitage, "Maximising Student Exposure to Unix Networking using FreeBSD Virtual Hosts," CAIA Technical Report 030320A, March 2003 http://caia.swin.edu.au/reports/030320A/CAIA-TR-030320A.pdf

[4] "VIA EPIA Mainboard ", (as at 28/7/20003), http://www.viavpsd.com/product/epia_mini_itx_spec.jsp?motherboardId=21

[5] "my.seagate.com: Barracuda 7200.7 - ST380011A", (as at 28/7/20003), http://www.seagate.com/cda/products/discsales/marketing/detail/0,1081,581,00.html

[6] "Intel 82559 Fast Ethernet Multifunction PCI/Cardbus Controller Datasheet", (as at 28/7/20003), http://www.intel.com/design/network/datashts/738259.htm

[7] "FreeBSD/i386 4.8-RELEASE Hardware Notes", (as at 28/7/20003), http://www.freebsd.org/releases/4.8R/hardware-i386.html

[8] David E. O'Brien , "FreeBSD Hypertext Man Pages: fxp(4)", (as at 28/7/20003), http://www.freebsd.org/cgi/man.cgi?query=fxp&sektion=4&manpath=FreeBSD+4.8-RELEASE

[9] "Swinburne Information Technology Services Home", Last Updated: Thursday, 12-Jun-2003 12:15:44 EST http://www.its.swin.edu.au/

[10] " Swinburne Information Technology Services – Networks", Last Updated: 22-May-2003 09:43:16, http://www.networks.swin.edu.au/services/wlan/index.php

[11] "CERT Coordination Center; Anonymous FTP Configuration Guidelines", (as at 28/7/20003), http://www.cert.org/tech_tips/anonymous_ftp_config.html

[12] "The Apache Software Foundation", (as at 28/7/20003), http://www.apache.org/

[13] "The Official Half-Life Web Site", (as at 28/7/20003), http://games.sierra.com/games/half-life/

[14] "GamePlanet Downloads: Half-Life: Official: Dedicated Server", (as at 28/7/20003), http://downloads.gameplanet.co.nz/cat.dyn/Half-Life/Official/Dedicated_Server/

[15] "The Linux Homepage at Linux Online", (as at 28/7/20003), http://www.linux.org/

[16] "IPFW HOWTO V.0.3", (as at 28/7/20003), http://www.freebsd-howto.com/HOWTO/Ipfw-HOWTO

[17] dragorn@kismetwireless.net, "KISMET: 802.11 wireless network sniffer", (as at 28/7/20003), http://www.kismetwireless.net/

[18] "Net Stumbler . com - Proudly Stumbling a Street Near You ", (as at 28/7/20003), http://www.netstumbler.com/