Adding and securing a Public Wireless Access Point within a home network

GSEC Practical Assignment - Version 1.4c – Option 2

Author: Steven Christall

Date: 08 December 2004

Abstract

This project details the migration of a simple home wireless network to include a public wireless access point. This is done using open source products and utilising older, retired hardware. Possible solutions for the network layout and security risks are evaluated, along with the risks of supplying public services. A solution is chosen and the steps to implement, secure and test the security of the final solution are described.

Table of Contents

Background on Public Wireless Access Points

There are a large number of solutions available to provide Public Wireless Access Points[A]. Some are commercial end to end solutions providing billing, wireless connectivity, and actual Internet connectivity. Other solutions only provide the Captive Portal and traffic management functions.

A good description of what a Captive Portal attempts to accomplish is quoted below[B]

> "Captive portals allow you to leverage a common browser as a secure authentication device."
>
> "Captive portals are becoming a popular way" ..... "to provide user authentication and IP flow management (basically traffic shaping and bandwidth control) without a required client application. They work by forcing un-authenticated users to a web page, once you have 'captured them' this way by allowing the web page to interact with the router/firewall you can completely control their access."

In this case we want to provide a public access point with basic authentication and bandwidth monitoring. Since it is being offered as a public service we are not concerned with collecting money!

To this end the Wireless Access Point is based mostly on pre-owned hardware and implemented using low cost or free open source software.

Project Requirements

In order to provide a Public Access Point from a home network, the following criteria have been deemed as essential.

- Security – the addition of a Public Access Point should in no way subtract from the security of the existing home network

- Bandwidth – the amount of bandwidth available to the Public Access point should be at a minimum trackable and preferably controllable.

- Access control – Initially the aim is to provide simple easy access to the Internet to anyone within range of the Public Access point. Eventually it would be preferable to require an email-style registration where any new user has to submit their email address and then gains five minutes of Internet access to pickup their mail and obtain a verification token. This would at least give some kind of traceability / contact-ability to a user if required. Initially, however, this requirement will be considered out of scope for this paper, as there are a large number of other requirements to be met.

Available hardware

A large amount of older hardware was available to dedicate to this project without any cost. The available hardware includes various PCMCIA network cards, both wired and wireless with external antenna jacks, as well as several older laptops to use as gateways / firewalls.

Available software

- Captive Portals

After some research it became apparent that there were several purpose built Wireless Access Point based distributions available[C]. However on closer examination they were determined to be too restrictive to fit into the existing network architecture.

For this project two open source Captive Portal solutions were considered.

1 / NoCatAuth[D] was developed by a Wireless Community group in Sonoma County, California. NoCatAuth is based around a series of perl scripts that dynamically control firewall rules and hence access. A new successor to NoCatAuth, named NoCatSplash[E] is currently being developed in C, with an aim to reduce its dependencies and size, in turn becoming more suitable for imbedded devices. As of the writing of this paper NoCatSplash was missing a large proportion of the functionality of NoCatAuth.

2 / Chillispot[F] is actively being developed and maintained by Jens Jakobsen, with contributions from others. It is a more complete implementation than NoCatAuth and is primarily written in (ANSI) C. It supports advanced features[G] such as authentication via Wifi Protected Access (WPA), authentication with Radius servers, as well as reporting functionality. It also supports a universal logon method, based on a “Captive” redirect to a cgi based webpage.

- Firewalls / Gateways

There are a large number of Linux based firewall distributions available today. They include appliance-style installations like Smoothwall[H] or Astaro Firewall[I], where a hardened version of Linux is included as part of the install. Both of these firewall products offer a free open source version to the Home / Soho user, while selling a commercial version targeted at larger companies.

Other options include the use of ruleset builders such as Firewall Builder[J], which through the use of “policy compilers” supports any version of Linux running iptables, as well as ipfilter and OpenBSD PF.

Existing network

The existing network was a relatively simple design, with a Draytek Vigor 2600 Asymmetric Digital Subscriber Line (ADSL) router providing connectivity to the Internet via many to one Network Address Translation (NAT)^{K}. There was no demilitarized zone (DMZ)^{L} configured nor was it available on this device. In the private[1] network there was a wireless bridge servicing laptops with 128 bit Wired Equivalent Privacy (WEP)^{M} encryption, and several wired Ethernet clients including a Media server and a Web server. See figure 1 below.

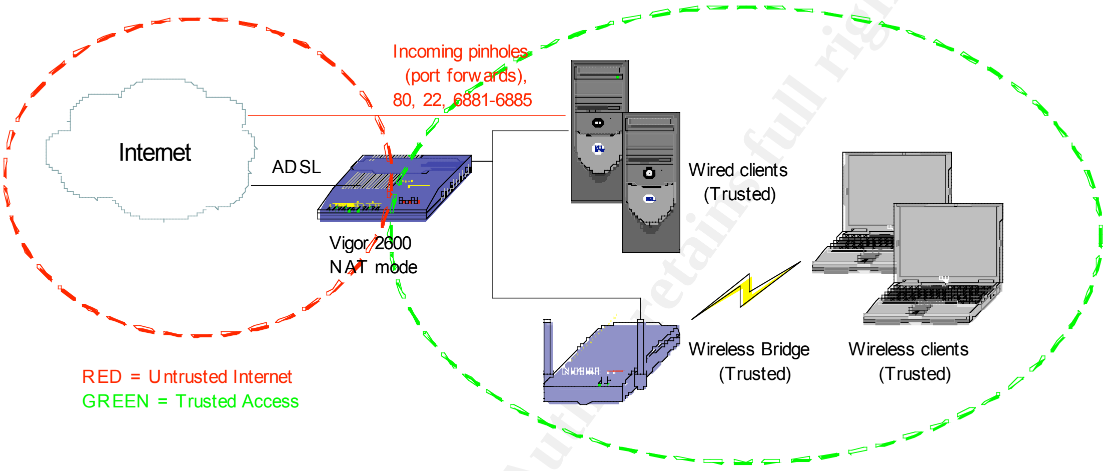


Figure 1 – Existing Network Architecture

The Draytek router provides a relatively full-featured firewall, which was not used other than discarding any inbound and outbound Microsoft NetBIOS traffic. Since the router was being used in NAT mode, any incoming traffic to the external interface is by definition denied. In addition there are several pinholes punched through providing incoming connections for SSH, HTTP, and Bittorrent to various clients in the trusted network.

Existing security risks

There is some risk associated in providing access to selected services on clients in any trusted network. In a more commercial setting, these services would typically be segregated onto hardened hosts running in a DMZ. However in the case of this home network the cost compared to the risk was deemed too high.

Additionally the existing wireless bridge only supports WEP encryption. This is based on the RC4 encryption protocol, utilising a shared secret. This was

[1] For the rest of this paper, the internal private network will be referred to as the Trusted network.

best encryption available when the bridge was purchased in 2002, however studies have highlighted weaknesses[N] in the implementation of this algorithm and tools[O] are available to detect the shared secret. In an attempt to reduce the risk the author manually re-enters the shared secret based on 26 completely random hex numbers on the wireless devices every three months.

While it is acknowledged that there is a risk that a determined hacker could gain access to the trusted wireless network, the manual rotation of the keys, combined with low data volumes reduce this risk. In the near future the author plans to replace all of the 802.11b wireless equipment with 802.11g based equipment, which includes the newer and less vulnerable Wifi Protected Access (WPA) encryption.

Possible network solutions

Solution 1

Unfortunately the Draytek Vigor 2600 does not provide DMZ functionality. Ideally the Public Wireless Access Point[2] would be segregated in its own network. The Draytek does however support running in a non-NAT mode, essentially acting as an ADSL router. In order to run in this mode, and to secure both the Trusted network and the Public network, it would require a minimum of two, or even three usable IP's from the ISP. Ideally one IP for the router, one for the Trusted network NAT device and the third for the Public network NAT device.

Figure 2 below details this network.

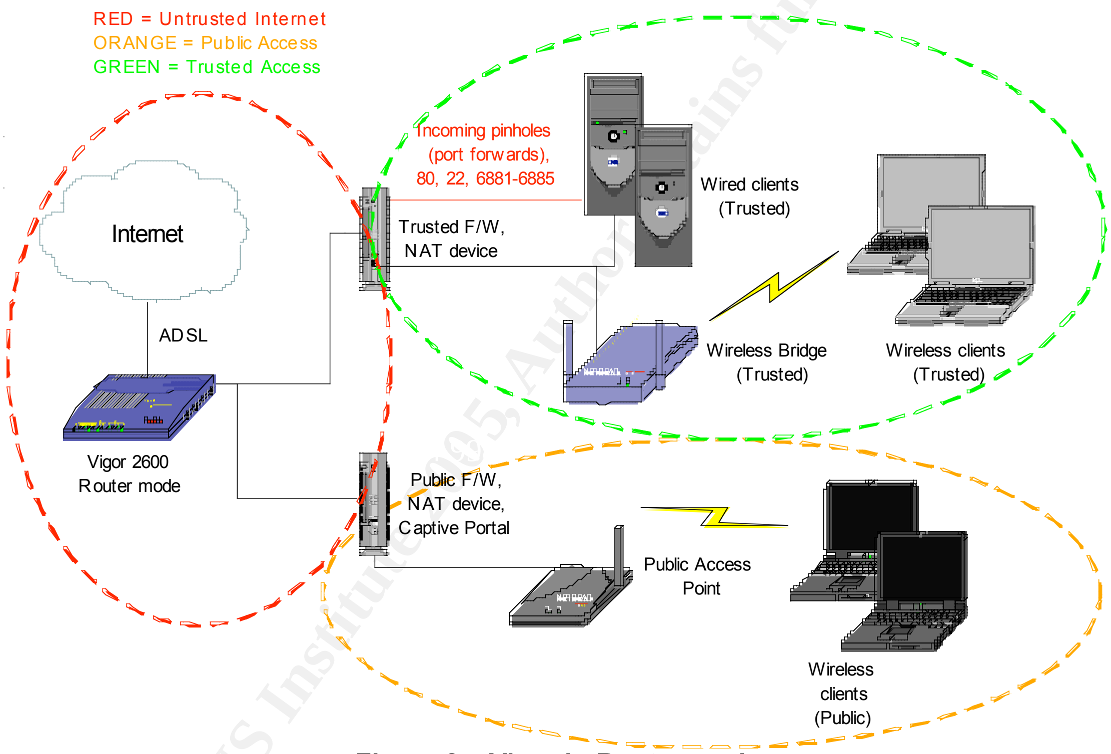


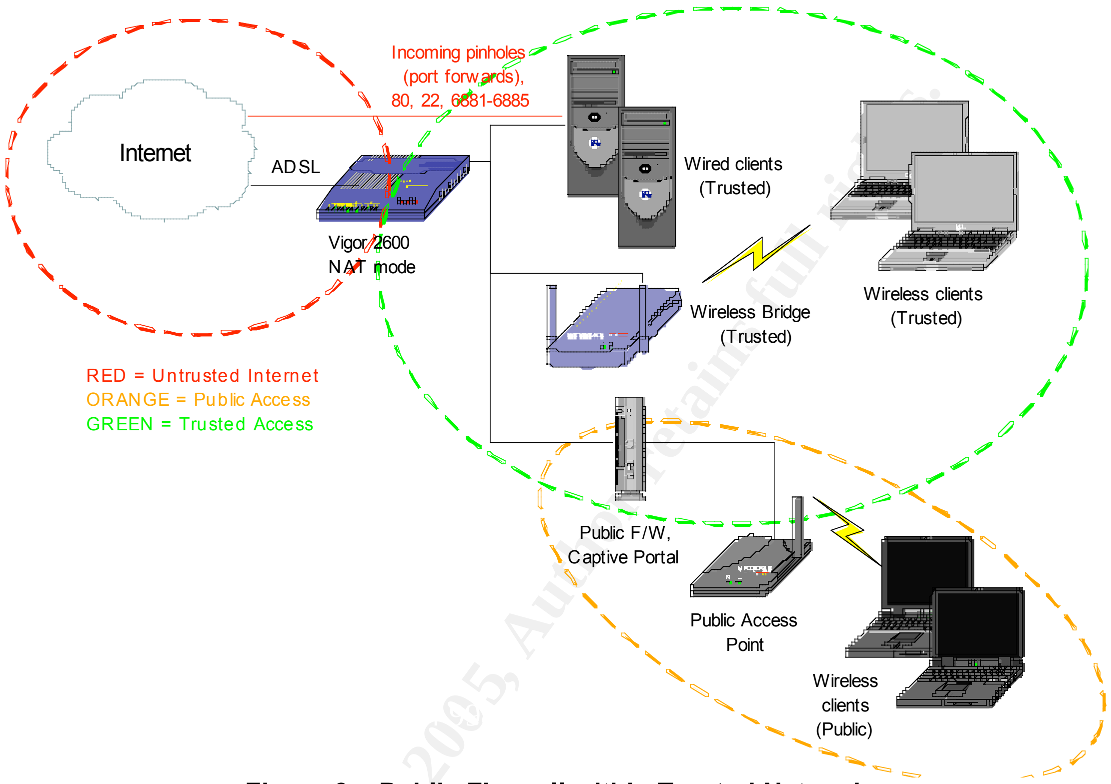
Figure 2 – Vigor in Router mode

This solution provides the highest level of separation of the Trusted and Public networks. By having two separate firewalls there is a little or no chance of changes on one affecting the other. In addition some Captive Portal implementations dynamically rewrite the firewall ruleset on the hosting server, which from a security perspective increases the complexity and risk, especially if the Captive Portal was running on the firewall device protecting the Trusted network.

[2] For the rest of this paper, the Public network is deemed to be the network containing the Public Wireless Access Point and any associated devices, such as a router and Captive Portal.

Solution 2

For this solution the existing Draytek router is used in its current NAT configuration. The Public network's firewall / Captive Portal is placed within the Trusted network. By using stringent firewall rules on the Public firewall hosting the Public Access Point, we can segregate it from the rest of the Trusted network. Figure 3 below details this network.



Incoming pinholes (port forwards), 80, 22, 6881-6885
Internet
ADSL
Vigor 2600 NAT mode
Wired clients (Trusted)
Wireless clients (Trusted)
Wireless Bridge (Trusted)
RED = Untrusted Internet
ORANGE = Public Access
GREEN = Trusted Access
Public F/W, Captive Portal
Public Access Point
Wireless clients (Public)

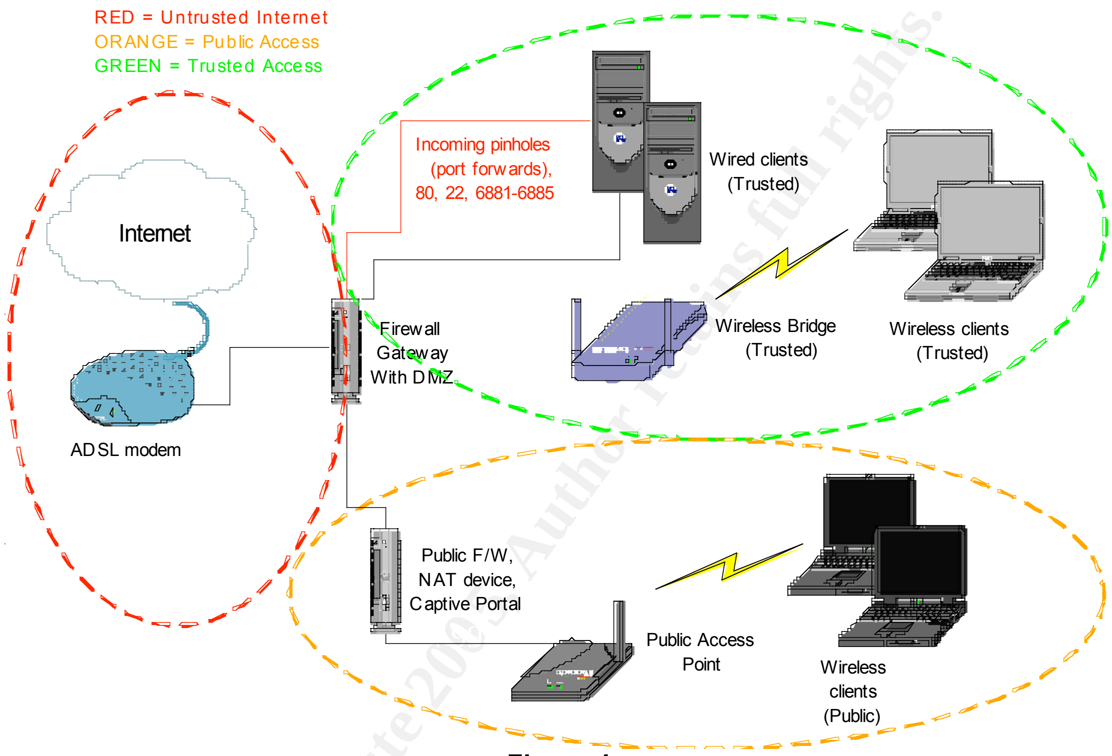
Figure 3 – Public Firewall within Trusted Network

This solution reduces the changes to be made to the existing network. Additionally it reduces the device count compared to solution one where we require two additional firewall / gateway devices. In this case we only require one.

Unfortunately this solution has large security concerns, as we are one hundred percent reliant on the firewall configuration of the Public firewall / gateway. If a Public network user were able to bypass the firewall on the gateway device attached to the Trusted network, either through nefarious means, or perhaps because of a simple misconfiguration of the firewall ruleset, they would have unfettered access to the Trusted network.

Solution 3

In this final suggested solution we discard the Draytek router and replace it with another ADSL firewall / router. In order to improve the network layout this device is deemed to have a DMZ, so we can have a centralized single firewall / gateway connecting to the Internet. We again use a second firewall / gateway device that also runs the Captive Portal software, in this case however it is segregated in the DMZ, providing additional protection if there are any issues with the firewall ruleset on the Public firewall.



RED = Untrusted Internet
ORANGE = Public Access
GREEN = Trusted Access
Internet
ADSL modem
Incoming pinholes (port forwards), 80, 22, 6881-6885
Wired clients (Trusted)
Firewall Gateway With DMZ
Wireless Bridge (Trusted)
Wireless clients (Trusted)
Public F/W, NAT device, Captive Portal
Public Access Point
Wireless clients (Public)

Figure 4

This solution requires a large number of changes to the existing infrastructure, however it is better structured, with a centralised point of access to the Internet. Additionally it provides Public clients with two independent layers of firewalls to pass through before gaining possible access to either the Trusted network or the Internet, which follows every security professionals mantra of Security in Depth.

Deciding what to use

The three suggested network solutions could be summarised as follows

	Solution 1 Vigor in router mode	**Solution 2 Public in Trusted**	**Solution 3 Replace Vigor**
Device count	Two additional firewall / router devices + Wireless access point	One additional firewall / router device + Wireless access point	Two additional firewall / router devices + Wireless access point + ADSL modem
Ongoing costs	Monthly fee for IP block. Maintenance of security updates x2 devices	Maintenance of security updates x1 device	Maintenance of security updates x2 devices
Security	Trusted network has similar level of security	Reduction of security in Trusted network	Possible increase in Trusted security; move live hosts into true DMZ

As stated in the original Project Requirements, the implementation of this project should in no way detract from the existing security of the Trusted network. This eliminates Solution 2. Solution 1 and 3 both use similar amounts of additional hardware, however Solution 1 has additional ongoing costs, while Solution 3 has additional possible benefits to the security of the Trusted network.

Solution 3, where the Vigor router is replaced by a firewall / router with a true DMZ is therefore chosen.

Next a firewall package needed to be selected for the replacement firewall / ADSL router. Since one of the constraints of this project is to implement it utilizing pre owned hardware, the selected firewall needed to be able to:

1 Run on laptop hardware with PCMCIA devices
2 Control an USB ADSL modem

Unfortunately this eliminated Astaro Internet Security, as it did not support laptops or ADSL devices. This left Smoothwall Express or a Redhat Linux distribution (as the Open Source OS the author is most familiar with) with a firewall policy written by Firewall Builder.

Ultimately this lead to the decision to use Smoothwall Express, as it is a massive task in itself to harden any installation of an OS, which would be a project in itself!

Finally a Captive Portal solution needed to be chosen. Earlier two possible packages were identified. ChilliSpot was chosen over NoCatAuth simply because it was in active development, has better documentation and the author was able to get it to work reliability.

The Chosen Solution

Implementation of Trusted Firewall / ADSL router

Smoothwall Express v2.0 was downloaded, md5sum verified and burnt to CD. It was very easy to install, however one should be aware that it will destroy all existing data on the computer on which it is being deployed. It does not offer any partitioning / data preservation options. This is inline with its appliance style of implementation.

A Pentium III laptop with one built in Ethernet and two PC card slots was chosen, giving a maximum of three "interfaces" available to partition networks, or even more including USB based devices

When installing Smoothwall it is initially configured locally on the machines console, utilising a menu driven, text based installer. Once the network interfaces have been detected, they are assigned to different firewall interfaces. Smoothwall uses the concepts of Red, Orange and Green interfaces, which map to Untrusted (Internet), Public, and Trusted networks respectively.

Once the Trusted interface has been configured and the management passwords set, Smoothwall continues to boot into a running state. The configuration of actual services, firewall rules and ADSL settings are carried out using a web-browser connecting to HTTP/HTTPS on ports 81/441.

Smoothwall Express has an impressive feature set which includes traffic graphs, DHCP server, DNS proxy server, Web proxy server, IDS server, VPN connections and more.

Out of the box Smoothwall Express is running in NAT mode, and hence by definition blocks all incoming connections from the Untrusted interface.

Smoothwall.org also produce patches for this open source product. These are provided to download in the form of a fixes patch file, which can then be uploaded and installed via the Web management interface (only available on the Green or Trusted network). This means that SmoothWall can be built and patched, even before it is connected to the Internet. Currently there are four patches available and all must be installed in sequence, one after the other. The last patch updates the running Kernel to 2.4.27

One issue that other users could have installing Smoothwall on a laptop is that the provided Kernel does not natively support PCMCIA devices. Luckily there is a very active community[P] supporting various modifications. One of these modifications is named "SuperKernel"[Q] which is essentially a recompiled fixes4 patch so that the included Kernel supports PCMCIA devices.

Finally one other patch was applied; “Full Firewall Control”[R] which modifies the firewall scripts to include rules for both incoming and outgoing connections. The default installation of Smoothwall Express only allows rules to block / manipulate incoming traffic.

Network Layout of Smoothwall Express Firewall

Smoothwall Interface Name	**Network name**	**Device**	**IP Address**
Green	Trusted	eth0 (Built in)	192.168.0.1
Orange	Public	eth1 (PCMCIA)	192.168.200.1
Red	Untrusted (Internet)	ppp0 (USB ADSL modem)	External address

Implementation of Captive Portal / Wireless Access point

Any Wireless Captive Portal requires hardware to support the Wireless Access Point. One of the objectives of this project was to use existing hardware where available. In this case there are two PCMCIA wireless cards available. Both cards have external antenna connections. One is based on the Prism2 chipset and the other on the Lucent/Agere chipset. Ideally we want to setup the wireless card in Access Point mode (Infrastructure, or Master mode), rather than Computer – Computer (Adhoc) mode. This is simply because most wireless clients will automatically connect to an access point advertising itself running in Master mode.

As there is a Linux driver named hostap[S] for Prism based chipsets that supports Infrastructure or Master mode, the Prism2 based card was used. This card is also more suitable as it is a high-powered (200mW) card, which when using an external antenna is advantages to offset cable and connector losses.

Again a Pentium III laptop with one built in Ethernet and two PC card slots was utilised, giving a maximum of three “interfaces” available to partition networks.

In this case we only need two networks to support the Captive Portal. One to connect to the Public network for Internet access and the other to connect to the Wireless Access Point. For this implementation the Wireless Access Point was provided by the operating system, rather than a physical Ethernet connection to an external Wireless bridge.

Since the author has used RedHat based distributions the most, RedHat Fedora Core2[T] is used to base the Public Wireless Access Point on. This is based on the 2.6 kernel (2.6.8 branch), which is now entering mainstream usage.

A standard server installation was performed and then a full yum[3] update run.

```
yum update
```

This updated all of the installed packages against Fedoras active update list.

[3] yum (Yellow Dog Updater, Modified), comes pre-installed with Fedora Core2. It provides dependency resolution for rpm based systems. http://www.linux.duke.edu/projects/yum/

Installing and configuring: Wireless Access Point driver

The hostap driver was then downloaded and installed, again using yum. The packages installed were

```
kernel-module-hostap-2.6.8-1.521
hostap-driver 0.2.5-7.rhfc2.at
hostap-utils 0.2.4-3.rhfc2.at
```

Upon insertion of the Prism based card, the hostap driver was automatically loaded. The card needs to be set into Master mode to use as an Access Point. This was done using iwconfig

```
iwconfig wlan0 mode master
```

We can also set the ESSID of the base station

```
iwconfig wlan0 essid FreeInternet
```

Since we want these settings to be set with each reboot, we need to add them as parameters to the /etc/pcmcia/hostap_cs.conf file

```
module "hostap_cs" opts "channel=11 iw_mode=3 essid=FreeInternet”
```

Here iw_mode=3 is Master mode, and we are choosing channel 11 so that this Access Point is at least 5 channels away from the Trusted Access Point reducing interference between the base stations[U].

We do not need to set an IP address for the wireless interface wlan0, as the Chillispot daemon takes care of this.

Installing and configuring: Captive Portal Software

As discussed previously Chillispot was chosen as the Captive Portal Solution.

This was again installed using yum. Chillispot also requires a Radius server to authenticate against. As discussed in the Chillispot installation notes[V], one possible option is to use FreeRadius[W].

The packages installed were

```
chillispot 0.98-1
freeradius 1.0.1-0.FC2
```

Next the /etc/chilli.conf file was modified as per the single machine configuration example[X]. In this case the radiussecret has been modified, as well as changing the DHCP client interface to match our Wireless Access Point. Finally a shared secret for the Universal Access Method (uam) authentication script has been added.

```
radiusserver1 127.0.0.1
radiusserver2 127.0.0.1
radiussecret ourradiussecret
uamserver https://192.168.182.1/cgi-bin/hotspotlogin.cgi
dhcpif wlan0
uamsecret ouruamsecret
```

Next the shared secrets were configured for both the radius server, as well as the uam logon script hotspotlogin.cgi

In /etc/raddb/clients.conf

```
secret  = ourradiussecret
```

was added immediately below the line starting with “client 127.0.0.1 {“

In /var/www/cgi-bin/hotspotlogin.cgi the line starting with #$uamsecret was edited to;

```
$uamsecret = "ouruamsecret”
```

Additionally the following text was added below the line starting with “<h1 style=\"text-align: center;\">ChilliSpot Login</h1>”

> “This is a free community based hotspot. You are able to access the Internet with no charge. Please enter guest as your username and password”

Finally a guest user was created in the radius database for our “guests” to authenticate against. By default FreeRadius queries a local user file /etc/raddb/users. The following line was added;

```
guest   Auth-Type := Local, User-Password == "guest"
```

Securing the Public Firewall / Access Point

Before starting the ChilliSpot Captive Portal, steps needed to be taken to secure the interfaces on the Public Firewall / Access Point that ChilliSpot is running on. There is an example firewall script provided with ChilliSpot package named firewall.iptables[Y]. This was modified to use our wlan0 card as the internal interface as well as adding a line to block outgoing SMTP connections from clients.

```
$IPTABLES -t nat -A PREROUTING -i tun0 -p tcp --dport 25 -j DROP
```

Other than this restriction to outgoing SMTP, once a client has authenticated, they have full access to the Internet.

At this point the Captive Portal was available to test. From the command line both of the daemons were started.

```
service start chilli
service start radiusd
```

Additionally to ensure that they start with each reboot

```
chkconfig chilli on
chkconfig radiusd on
```

Issuing an ifconfig shows a new interface

```
tun0    Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
        inet addr:192.168.182.1  P-t-P:192.168.182.0  Mask:255.255.255.0
        UP POINTOPOINT RUNNING  MTU:1500  Metric:1
        RX packets:634 errors:0 dropped:0 overruns:0 frame:0
        TX packets:576 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:10
        RX bytes:54499 (53.2 Kb)  TX bytes:383400 (374.4 Kb)
```

This is a virtual interface that the Chillispot daemon has created using the Linux tun[Z] driver (which is now included as part of the Linux Kernel distribution), and provides a captive network for clients attaching to our Wireless Access Point. In the author's opinion this is a more elegant and secure method for capturing and redirecting clients accessing the Captive Portal, as opposed to the dynamic firewall rule rewrites that NoCatAuth Captive Portal implements for example.

Network Layout of Public Firewall / Captive Portal

Network name	Device	IP Address
Public LAN	eth0 (PCMCIA)	192.168.200.20
Public Wireless	wlan0 (PCMCIA)	none
ChilliSpot Captive Portal	tun0 (Virtual)	192.168.182.1

Testing the Live System

A Windows XP laptop with built-in wireless connectivity was used to test the Public Wireless Access Point.

Windows XP immediately located and associated with the Public Wireless Access Point. The network configuration obtained via DHCP was;

ipconfig /all

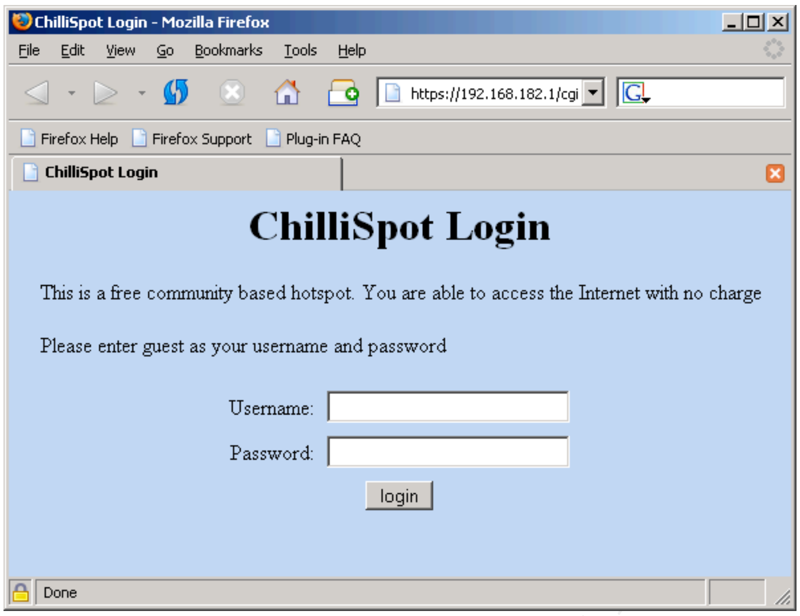
```
Ethernet adapter Wireless Network Connection:

    Connection-specific DNS Suffix  . : key.chillispot.org
    Description . . . . . . . . . . . : Dell Wireless WLAN 1350 WLAN Mini-PCI Card
    Physical Address. . . . . . . . . : 00-90-96-A8-C3-8D
    Dhcp Enabled. . . . . . . . . . . : Yes
    Autoconfiguration Enabled . . . . : Yes
    IP Address. . . . . . . . . . . . : 192.168.182.4
    Subnet Mask . . . . . . . . . . . : 255.255.255.0
    Default Gateway . . . . . . . . . : 192.168.182.1
    DHCP Server . . . . . . . . . . . : 192.168.182.1
    DNS Servers . . . . . . . . . . . : xxx.xxx.xxx.xxx
                                        xxx.xxx.xxx.xxx
    Lease Obtained. . . . . . . . . . : 07 December 2004 22:26:02
    Lease Expires . . . . . . . . . . : 07 December 2004 22:36:02
```

Note the short DHCP lease granted by the ChilliSpot Portal. This is to automatically detect clients that have switched off or moved out of range of the Captive Portal without logging off.

When opening a browser on the testing laptop, it was automatically redirected to the following URL

```
https://192.168.182.1/cgi-bin/hotspotlogin.cgi?res=notyet&uamip=192.168.182.1&uamport=3990&challenge=ff003c1b6457219f8637a617f1fe6d5e&userurl=http%3a%2f%2fslashdot.org%2f&nasid=nas01
```

This displayed the modified ChilliSpot login page with instructions for guest users shown in figure 5 below.


ChilliSpot Login - Mozilla Firefox

https://192.168.182.1/cgi

ChilliSpot Login

This is a free community based hotspot. You are able to access the Internet with no charge

Please enter guest as your username and password

Username:

Password:

Figure 5 - Wireless Portal Login

Entering a username and password of guest and clicking on login allowed the browser to continue to the originally requested page.

At this point the user has full access to the Internet with almost any application capable of surviving NAT. A popup window is opened with a link to logout, as well as a timer running to show the user how long they have been online.

Testing the Security of Implemented Solution

In the previous sections two Linux based Firewall / Router / Gateway devices were installed and configured. As a sanity check and general good practice after making any network changes, port scans were carried out.

1 / External port scans were made against the SmoothWall Express Firewall from a remote hardened host on the Internet.[4] The standard port range was increased to cover all ports for this external scan.

```
[root@ux-tubweb root]# nmap -sV -P0 -p1-65535 external.address.net

Starting nmap 3.48 ( http://www.insecure.org/nmap/ ) at 2004-12-07 19:21 GMT
Interesting ports on net.address.external (xxx.xxx.xxx.xxx):
(The 65531 ports scanned but not shown below are in state: filtered)
PORT     STATE  SERVICE       VERSION
22/tcp   open   ssh           OpenSSH xxxx (protocol 2.0)
113/tcp  closed auth
3390/tcp open   microsoft-rdp Microsoft Terminal Service (Windows 2000 Server)
6883/tcp closed unknown

Nmap run completed -- 1 IP address (1 host up) scanned in 2796.735 seconds
```

This is an expected result. Port 22 is being forwarded to a Linux host, and port 3390 is being forwarded to a Windows host. Port 6883 is part of a forwarded range to a client running Bittorrent.

2 / Internal port scans were also made against the SmoothWall firewall in both internal networks, as well as a scan made by a Public Wireless client prior to, and after registration.

- ChilliSpot gateway interface, prior to authentication

```
root@localhost ~ # nmap -sV -P0 192.168.182.1

Starting nmap 3.70 ( http://www.insecure.org/nmap/ ) at 2004-12-07 22:52 GMT
Interesting ports on 192.168.182.1:
(The 1658 ports scanned but not shown below are in state: filtered)
PORT    STATE SERVICE  VERSION
80/tcp  open  http     Apache httpd 2.0.51 ((Fedora))
443/tcp open  ssl/http Apache httpd 2.0.51 ((Fedora))
MAC Address: 00:60:B3:79:15:C1 (Z-com)

Nmap run completed -- 1 IP address (1 host up) scanned in 31.496 seconds
```

[4] When the original port scan was carried out, the author lost external access to the Smoothwall Express firewall. On arriving home and checking the logs it was discovered that the built-in IDS system had blocked the scanning IP.

- ChilliSpot gateway interface, after authentication

```
root@localhost ~ # nmap -sV -P0 192.168.182.1

Starting nmap 3.70 ( http://www.insecure.org/nmap/ ) at 2004-12-07 23:01 GMT
Interesting ports on 192.168.182.1:
(The 1658 ports scanned but not shown below are in state: filtered)
PORT    STATE SERVICE  VERSION
80/tcp  open  http     Apache httpd 2.0.51 ((Fedora))
443/tcp open  ssl/http Apache httpd 2.0.51 ((Fedora))
MAC Address: 00:60:B3:79:15:C1 (Z-com)

Nmap run completed -- 1 IP address (1 host up) scanned in 31.984 seconds
```

Both of the ChilliSpot scans have shown the Captive Portal listening on ports 80 and 443. This is expected as it captures browser traffic both before and after authentication.

- Public gateway interface

```
[root@ux-surf raddb]# nmap -sV -P0 192.168.200.1

Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) at 2004-12-07 19:23 GMT
All 1659 scanned ports on 192.168.200.1 are: filtered

Nmap run completed -- 1 IP address (1 host up) scanned in 72.417 seconds
```

This is an expected result. While this is a NAT interface it does not have any listening services.

- Trusted gateway interface

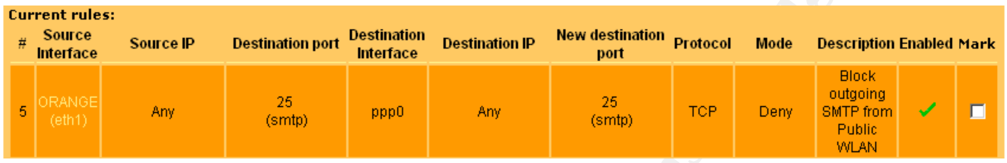
```
[root@ws root]# nmap -sV -P0 192.168.0.1

Starting nmap 3.48 ( http://www.insecure.org/nmap/ ) at 2004-12-08 00:20 GMT
Interesting ports on 192.168.0.1:
(The 1654 ports scanned but not shown below are in state: closed)
PORT    STATE SERVICE VERSION
81/tcp  open  http    Apache httpd 1.3.31 ((Unix) mod_ssl/2.8.19 OpenSSL/0.9.7d)
222/tcp open  ssh     OpenSSH 3.7.1p1 (protocol 1.99)
441/tcp open  http    Apache httpd 1.3.31 ((Unix) mod_ssl/2.8.19 OpenSSL/0.9.7d)

Nmap run completed -- 1 IP address (1 host up) scanned in 10.306 seconds
```

This is an expect result. The three management ports are available only on the Trusted interface

Layered Security

With the default installation of SmoothWall, all outgoing traffic is allowed. In order to better protect the Internet from Public Wireless users with free Internet access, an additional layer of security was introduced on the Internet facing firewall, ensuring that even if the ruleset on the Wireless Access Point is somehow circumvented, restrictions continued to apply.

This was achieved by adding a rule to the rule-set on the SmoothWall Firewall, blocking outgoing SMTP traffic from the Public (Orange) network to the Internet.



Current rules:

#	Source Interface	Source IP	Destination port	Destination Interface	Destination IP	New destination port	Protocol	Mode	Description	Enabled	Mark
5	ORANGE (eth1)	Any	25 (smtp)	ppp0	Any	25 (smtp)	TCP	Deny	Block outgoing SMTP from Public WLAN	✓	☐

Figure 6 - SmoothWall SMTP restriction

Summary / Conclusions

This project has been a very good learning exercise for the author and in some ways quite daunting when one looks at the amount of knowledge and research required to achieve what originally seemed a relatively simple task.

It became very obvious from the start that there are a large number of more than capable open source tools to perform almost any task, including Captive Portals, if the time is taken to research and locate them.

Here a basic installation of a Public Wireless Access Point has been achieved in a home network without detracting from the security of the original network. It has also been achieved utilising open source products and older low cost hardware.

Traffic graphs detailing the usage of the Wireless Access Point are available in the web interface of the SmoothWall Express Firewall.

While more advanced features such as bandwidth reporting and actual user verification through an email based system have not yet been implemented, it is the authors intention to do this and more.

References

[A] PersonalTelco. Portal Software. 07 Nov 04
<http://wiki.personaltelco.net/index.cgi/PortalSoftware>

[B] Shand Adam. Captive Portal. 07 Nov 04
<http://wiki.personaltelco.net/index.cgi/CaptivePortal>

[C] NYC Wireless. Pebble 2001-2005. Oct 04
<http://www.nycwireless.net/pebble/>

[D] Sonoma County Wireless Community. NoCatAuth. 8 Nov 04
<http://nocat.net/>

[E] Sonoma County Wireless Community. NoCatSplash. 28 Nov 04
<http://nocat.net/download/NoCatSplash/>

[F] Mondru AB. ChilliSpot Home Nov 2004
<http://www.chillispot.org/>

[G] Mondru AB. ChilliSpot Features 27 Sept 04. 25 Nov 04
<http://www.chillispot.org/features.html>

[H] The SmoothWall Open Source Project. SmoothWall Express 2000-2004. Sept 2004
<http://www.smoothwall.org/>

[I] Astaro AG. Astaro.org Site 2003-2004. Sept 2004
<http://www.astaro.org>

[J] NetCitadel, LLC. Firewall Builder 25 Oct 2004. Sept 2004
<http://www.fwbuilder.org/>

[K] Viacomsoft. Network Address Translation FAQ. Oct 2004
<http://www.vicomsoft.com/knowledge/reference/nat.html>

[L] Robertson, Paul D, Matt Curtin, and Marcus J. Ranum. Internet Firewalls: Frequently Asked Questions 26 Jul 04. Oct 2004
<http://www.interhack.net/pubs/fwfaq/firewalls-faq.html#SECTION00048000000000000000>

[M] searchSecurity.com Definitions. Wired Equivalent Privacy 15 Sept 04
<http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci549087,00.html>

[N] Fluhrer, Scott, Itsik Mantin, and Adi Shamir.
Weaknesses in the Key Scheduling Algorithm of RC4 2001. Oct 04
<http://www.drizzle.com/~aboba/IEEE/rc4_ksaproc.pdf#search='weaknesses%20in%20key%20scheduling%20algorithm%20rc4'>

[O] WEP cracking tools
Rager, Anton T. Wepcrack Oct 04 <http://wepcrack.sourceforge.net/>
Snax. AirSnort 22 Sept 04. Oct 04 <http://airsnort.shmoo.com/>

[P] mpot. Smoothwall Express community mods 13 Dec 03. Oct 04
<http://community.smoothwall.org/forum/viewtopic.php?t=2873>

[Q] BoHiCa. [2.0 fixes4] Super-Kernel 2.4.27-v2 19 Oct 04. Oct 04
<http://community.smoothwall.org/forum/viewtopic.php?t=7848&highlight=superkernel>

[R] Netwiz. Full Firewall Control v.1.1.5 - Firewall Access Control 17 Nov 04.
<http://community.smoothwall.org/forum/viewtopic.php?t=9593>
<http://sourceforge.net/project/showfiles.php?group_id=114890&package_id=132134>

[S] Malinen, Jouni. Host AP driver for Intersil Prism2/2.5/3.
<http://hostap.epitest.fi/>

[T] Redhat Community, Redhat Fedora 2004. Oct 2004
<http://fedora.redhat.com/download/>

[U] Higgins, Tim. When Wireless LANs collide 28 Feb 04. 28 Nov 04
<http://www.tomsnetworking.com/Sections-article75-page3.php>

[V] Mondru AB. ChilliSpot 0.98 Release Notes. 25 Nov 04
<http://chillispot.org/release.html>

[W] The FreeRADIUS Project. FreeRADIUS. 25 Nov 04
<http://www.freeradius.org/>

[X] Jakobsen, Jens. Chillispot FAQ. 25 Nov 04
<http://www.chillispot.org/FAQ.html#mozTocId30443>

[Y] Mondru AB. ChilliSpot 0.98 Release Notes. 25 Nov 04
<http://chillispot.org/release.html#mozTocId560609>

[Z] Krasnyansky, Maxim. Universal TUN/TAP driver FAQ.
1999-2001, 01 Dec 04
<http://vtun.info/vtunsite/tun/faq.html#1.1>

Bibliography

Fllickenger, Rob. Wireless Hacks – 100 Industrial-Strength Tips and Tools. Sebastopol: O'Reilly Media, Inc, 2003

SANS Institute. Track 1 – SANS Security Essentials. Volume 1.1. SANS Press, Jan 2004