

# ***Securing a Wireless Network using a VPN***

Scott Thomas and Hugh Smith  
Department of Computer Science  
California Polytechnic State University  
{sbthomas, husmith}@calpoly.edu

## ***Abstract***

This article walks you through securing a wireless network. Our configuration is focused on securing wireless network traffic for a small network such as a home or dorm room. The security provided is both authentication, allowing only known authorized users access to your wireless network, and encryption, preventing anyone from reading your wireless traffic. The solution we describe utilizes the open source FreeS/WAN software that implements the Internet Protocol Security (IPSEC) protocol. In addition to your wireless components (wireless NIC in your PC and wireless access point) you will need a machine running Linux to act as your security gateway. While our configurations assumes your wireless PC clients are running Linux, Windows 2000/XP based machines come equipped with VPN software which will allow you to interface with this configuration.

## ***Introduction***

Wireless is here! The Wi-Fi rush has begun and there are access points popping up all over the globe. This new craze, based on the 802.11b standard, is the new hit among students and professionals everywhere. The ease of use and speed is appealing to all computer users. Large and small conferences everywhere are now advertising free, open wireless access to all attendees. Many conferences will even check out wireless PCMCIA cards to use in your personal laptop. Unfortunately, wireless access is easier to spread than contain. Anyone with a wireless access card can see the traffic and read the data. There have been some advances in wireless security, namely WEP (Wired Equivalency protocol), however these protocols have been proved to be less than adequate [Fluhrer-2001]. We all wince as we send off our plain text passwords, hoping no one out there will notice.

So, what can we use to secure our wireless LANs from outsiders? We want a secure, private link to our access point so that no one else can read our data. Our suggestion is to implement a Virtual Private Network (VPN) over this insecure wireless link. VPNs establish an encrypted connection for traffic to pass through between the client and server. VPNs also offer authentication to ensure that only registered users use the wireless access point. This paper provides an easy to implement, scalable, and free solution using VPNs to secure wireless communications.

## ***Background***

### **Wireless Networks**

A wireless network in this paper refers to one or more computers, communicating via the 802.11b standard in infrastructure (access point) mode. Each client must be MAC level authenticated and associated to the access point. Once this process has taken place, the user is part of the wireless network and can send data to others on this network.

Wireless networks have many advantages over wired networks. First, and most obvious, is the fact that there are no wires to run. Cabling costs accumulate rapidly and degrade the visual appearance of a home or office. Wireless access points are also fairly cheap and most come setup right out of the box. The ease of use and affordability make this device a top choice among computing enthusiasts. Also contributing to the advantages of wireless networks is its speed. The 802.11b standard moves data at 11 MB/s, which is faster than most home Internet connections. However, a flailing red flag follows this standard wherever it goes. The encryption for 802.11b networks is the Wired Equivalency Privacy (WEP) protocol.

The main purpose of WEP is to make the wireless network as secure as possible by not allowing eavesdroppers to understand the user's data [Borisov – 2001]. It was designed to be as secure as a wired Ethernet connection, which provides no built in encryption. Since, the only security offered by wired networks is the physical denial of access to the network, WEP cannot be used as a means of guaranteeing secure data transmission. With the speed at which 802.11b networks are spreading, something must be done to secure data traffic. Many universities and companies are already implementing wireless LANs, both intentionally and unintentionally. Most wireless access points come out of the box with no encryption or restrictions on wireless access. This can create a huge hole into a companies or universities backbone.

### **Virtual Private Networks**

A virtual private network is a method of communicating data securely over an insecure network. A private connection is established between a VPN enabled client and VPN gateway and all data transmitted across this connection is encrypted. Anyone monitoring the traffic between the client and the gateway will see a standard IP packet header but the rest of the packet will be encrypted. The IPsec protocol is the standard protocol used to implement most VPNs as specified in RFC 2401 and RFC 2411.

There are two different modes of Virtual Private Networks: Transport mode and Tunnel mode. Transport mode is used when an application encrypts the traffic between a client and a server. An example of this is establishing a secure connection to a bank. Using tunnel mode means that any application not supporting encryption will send its traffic as

clear text. Since we wish to encrypt all traffic going across our insecure wireless link we will use our VPN in tunnel mode.

In tunnel mode two VPN gateways establish a tunnel for packets to travel through while they are going over a public network. (See figure 1.) The data originates on or somewhere behind the first VPN gateway, and finds its destination behind the other end of the tunnel. The VPN gateways encrypt the data before sending it across the insecure network. When receiving encrypted packets the gateway will decrypt the packet and then pass the packet to the machines behind them. The machines behind the VPN gateways are not aware that the data is being encrypted before being sent across the insecure network.

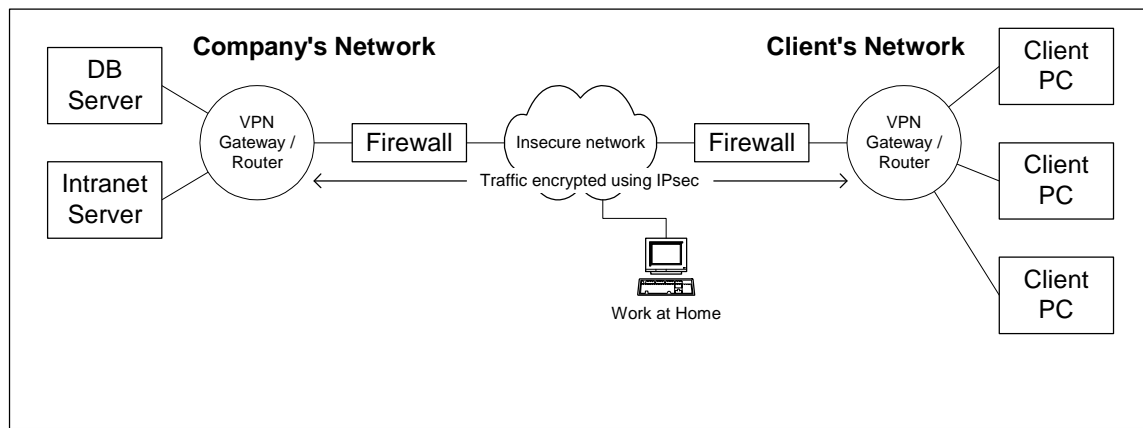


Figure 1: Common Industry VPN Implementation Using Tunnel Mode

There are many advantages of using VPNs. First the connection is fairly easy to set up, modify, and tear down. The alternative for most companies is to use a leased line from one office to another. This is secure, but can be extremely expensive, and is almost impossible to change if a branch relocates. VPNs not only provide a secure, encrypted link, but also offer authentication. This means people are who they say they are. However, all this security does not come for free. Encryption is not an easy task and can slow down the data transfer if sufficient hardware is not available. Also, configuration can be difficult since there are many encryption and authentication protocol parameters to choose from.

## ***Design and Implementation***

### **System Overview**

Figure 2 depicts the scenario we are working with. In our scenario we have a number of Linux based clients who wish to communication over an insecure wireless network. In addition to securing our data using encryption we wish to prevent unauthorized use of our wireless network. To simplify our configuration these clients obtain their IP

configuration (IP address, Default Gateway and DNS Server address) using DHCP. While our clients are Linux based there is a version of IPSEC that ships with Windows 2000/XP that will work on our VPN configuration.

The Linux based VPN gateway in figure 2 has two network interface cards. One interface is connected via a crossover cable to the wireless access point. The second interface is connected to the local area network with an Internet connection. On the wireless side, the VPN gateway will be responsible for assigning dynamic IP addresses to requesting clients. It will take requests to establish secure tunnels, and upon authentication route encrypted packets from the clients. The VPN gateway will decrypt these packets and send them on to the local wired LAN where they are routed across the Internet. Packets inbound for our VPN clients will arrive at the VPN gateway. The gateway receives these packets, encrypts them and transmits to the VPN clients via the wireless network. The client decrypts the packets and passes them on to the application.

In order to support this scenario the Linux machine must perform a number of functions. These are:

- 1) DHCP Server – The Linux machine is the DHCP server for all of the wireless clients.
- 2) DNS Server – The DNS configuration will be used to provide the IPSEC key lookups.
- 3) VPN Gateway – The machine will be running the VPN software and is responsible for encrypting/decrypting traffic across the wireless network.
- 4) Default Gateway/Router – The Linux machine will be configured to route traffic for the clients to the Internet.
- 5) Firewall – In order to prevent unauthorized users from accessing our network the Linux machine will be running firewall software (iptables).

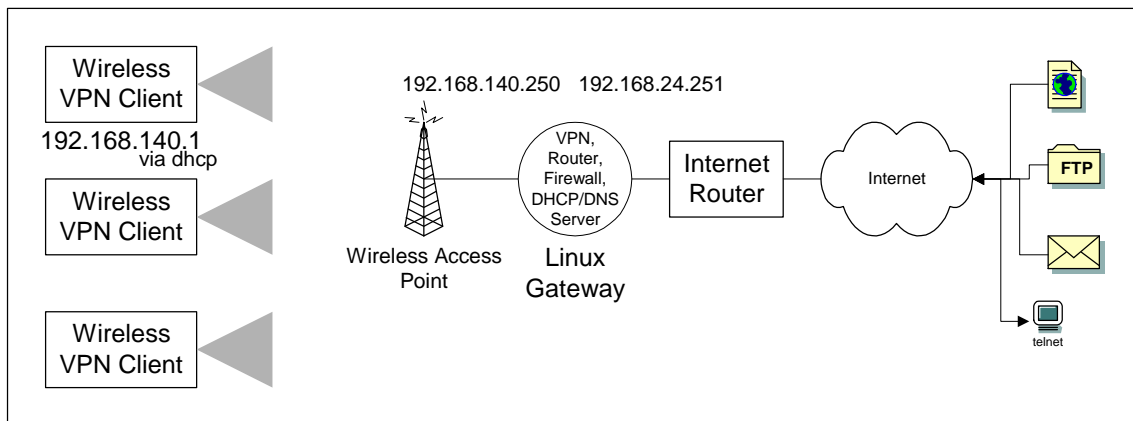


Figure 2: Securing Wireless Using a VPN

## **FreeS/WAN – VPN Software**

We decided to use FreeS/WAN (pronounced free-swan), an IPsec implementation for the Linux operating system. FreeS/WAN is freely available under the GPL license. The official web site is located at <http://www.freeswan.org>. This is an excellent web site and will be able to answer almost any question you may have regarding the FreeS/WAN package.

FreeS/WAN uses IPsec to encrypt traffic at the IP level of the protocol stack. This implementation uses Internet Key Exchange (IKE) as a way of securely exchanging keys. This exchange is based on the Diffie-Hellman key exchange protocol. The details of the exchange will not be covered here and can be found in RFC 2409. IKE uses the Internet Security Association Key Management Protocol (ISAKMP) to send the keys back and forth across the network. This protocol is detailed in RFC 2408. In order to assure authentication, Rivest Shamir Adleman (RSA) keys are used in this exchange. These keys provide much more security than shared passwords. The server will use the Domain Name Service (DNS) to authenticate the appropriate keys. Instead of a normal hostname lookup (type A), the server performs a key (type KEY) lookup to obtain a key for the client. Once a secure tunnel has been established, FreeS/WAN uses an Encapsulated Security Payload (ESP) to transmit data. The details of ESP can be found in RFC 2406. All these technologies combined together are used to create the encrypted tunnel for our data.

## **Scenario Configuration**

There are a number of steps to configuring both the gateway and the clients to support the scenario describe earlier. These steps are:

- 1) Install the FreeS/WAN software on both the client and gateway machines.
- 2) Configure DNS on the gateway machine. This configuration will include both machine names as well as the public keys needed for encryption.
- 3) Configure the gateway machine to be a DHCP server.
- 4) Set up the gateway's firewall and routing
- 5) Turn on the IPSEC services for your tunnel.

## **Installing FreeS/WAN**

The gateway machine we used is running Redhat Linux 7.3 with the 2.4.18-3 (default) kernel. The client machine is a laptop running Redhat Linux 7.2 with kernel 2.4.7-10 (default).

First download the two rpms appropriate to the PCs. There are not separate rpm for frees/WAN clients and the gateway. (Note: these files can be found on the FreeS/WAN website)

### Redhat 7.3 with 2.4.18-3 kernel

- `ftp://ftp.xs4all.nl/pub/crypto/freeswan/binaries/RedHat-RPMs/2.4.18-3/freeswan-module-1.98b_2.4.18_3-0.i386.rpm`
- `ftp://ftp.xs4all.nl/pub/crypto/freeswan/binaries/RedHat-RPMs/2.4.18-3/freeswan-1.98b_2.4.18_3-0.i386.rpm`

### Redhat 7.2 with 2.4.7-10 kernel

- `ftp://ftp.xs4all.nl/pub/crypto/freeswan/binaries/RedHat-RPMs2.4.7-10/freeswan-module-1.98b_2.4.7_10-0.i386.rpm`
- `ftp://ftp.xs4all.nl/pub/crypto/freeswan/binaries/RedHat-RPMs2.4.7-10/freeswan-1.98b_2.4.7_10-0.i386.rpm`

Install the two rpms you just downloaded on both machines:

- `rpm -ivh freeswan-module-*.rpm`
- `rpm -ivh freeswan-*.rpm`

Stop the IPSEC service from running on both PCs. (You will turn it back on later once you get the configuration files setup.)

- `service ipsec stop` (remember to do this on both the client and the gateway)

### Gateway DNS setup:

We now want to configure the DNS service on the gateway machine.

- On your gateway Linux machine edit the file `/etc/named.conf` and add the following lines:

```
zone "wireless.netprl.calpoly.edu" {
    type master;
    file "wireless.netprl.calpoly.edu.zone";
};
```

- Also on the gateway machine, create the file `/var/named/wireless.netprl.calpoly.edu.zone`

```
$TTL 86400
@      IN      SOA    @  root.localhost (
                        4 ; serial
                        28800 ; refresh
                        7200 ; retry
                        604800 ; expire
                        86400 ; ttl
                        )

@      IN      NS     gateway.wireless.netprl.calpoly.edu.

gateway      IN      A       192.168.140.250
```

You will now need to add an entry for each wireless client that will be using the VPN. To do this run the following command on each client machine: (This assumes you have installed FreeS/WAN on each client machine)

- `ipsec showhostkey`

This command will print out the following (The key has been shortened for clarity):

```
; RSA 2048 bits      localhost.localdomain      Sat Nov 2 13:53:22 2002
localhost.localdomain IN      KEY  0x4200 4 1 AQOVx0Jjl/ . . . 3rk6x
```

- Append these lines to `/var/named/wireless.netprl.calpoly.edu.zone` on the gateway machine. (It is probably easiest to save the output of the “`ipsec showhostkey`” command to a file on the client machine and then paste them into `/var/named/wireless.netprl.calpoly.edu.zone` on the gateway).
- Change `localhost.localdomain` to a unique identifier. Here is the example for a client1.

```
client1      IN      KEY      0x4200 4 1
AQOVx0Jjl/XwiMVdyea8oUdRXDauxpp14pekpRa+m5FEu0k4bOkL2PEQU83rXtKlPaN286zX
DJCH1/Ik3PGc2sgA+tuXJ3Syq4a21rqGBnibTJRcbF/zYQDRtLegHnURw3qqXep83F2/s0YL
toeGerc7McMJKAE7De8CerOb8vsg/h6jBPmxQH8kjo167CKdKTFm0uMQA fLNCZJcOyxJmcyh
295vwdFFrWoDb0yrhn9115DKP6+ZiRFI0IR6ItOOhe5WrhyMSE7aFf9YgczaG48HV/2eaLdo
xTQ0ciFQ+hIxRudh814oCSPEvvC/2Ot6+3PKsHNmjmyO3BdJvNR/lhcin+ijFuBIrVIv49HB
hWo3rk6x
```

Now start the DNS server on the gateway machine.

- `service named start`

Next edit the gateway machine’s file `/etc/resolv.conf` and verify it only contains the line:

```
nameserver 192.168.140.250
```

To test the DNS server

- `dig @192.168.140.250 client1.wireless.netprl.calpoly.edu -t KEY`

This should return the RSA key for the client you just entered. If not, something went wrong.

## Server DHCPD setup:

We now have to set up the DHCP server on the gateway. The DHCP server will support all clients in configuring their IP address, default gateway and DNS server. The client machines will be on the 192.168.140.0 subnet. This wireless subnet is named: wireless.netprl.calpoly.edu. Machines on this subnet are named client1, client2, etc. Notice that the DNS server for the clients is not the gateway machine. It is our normal DNS server used for Internet connection (in our case 10.10.10.254 and 10.10.11.254).

- On the gateway machine, edit /etc/dhcpd.conf to match the following figure:

```
default-lease-time 86400;
max-lease-time 604800;

option subnet-mask 255.255.255.0;
option broadcast-address 192.168.140.255;
option routers 192.168.140.250;
option domain-name "wireless.netprl.calpoly.edu";
option domain-name-servers 10.10.10.254, 10.10.11.254;

subnet 192.168.140.0 netmask 255.255.255.0 {
    range 192.168.140.1 192.168.140.20;
}
```

## Firewall / Routing setup:

Using iptables we will now configure the firewall on the gateway machine. Our firewall rules only allow routing of encapsulated security payload packets (IPSEC packets). A user logged into the Linux gateway machine itself will no longer be able to access the Internet. This can be fixed by modifying the script. However, be careful to not allow all traffic through.

- On the gateway machine, create the executable file /etc/rc.d/ipsec.firewall to match the following figure: (Remember to chmod 700 /etc/rc.d/ipsec.firewall when you are done to make the file executable.)

```

# Turn on IP forwarding
echo 1 > /proc/sys/net/ipv4/ip_forward

# Flush all chains
iptables -F

# Set default policies
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT

# Allow clients to request IP
iptables -A INPUT -p UDP -i eth0 --destination-port 67 -j ACCEPT
iptables -A INPUT -p UDP -i eth0 --destination-port 68 -j ACCEPT

# Allow DNS to localhost
iptables -A INPUT -p UDP --destination-port 53 -s 192.168.24.251 -d
192.168.24.251 -j ACCEPT
iptables -A INPUT -p UDP --source-port 53 -s 192.168.24.251 -d
192.168.24.251 -j ACCEPT
iptables -A INPUT -p UDP --destination-port 53 -s 192.168.140.250 -d
192.168.140.250 -j ACCEPT
iptables -A INPUT -p UDP --source-port 53 -s 192.168.140.250 -d
192.168.140.250 -j ACCEPT

# Allow Authentication
iptables -A INPUT -p UDP -i eth0 -s 192.168.140.0/24 --destination-
port 500 -j ACCEPT

# Allow Encrypted data
iptables -A INPUT -p ESP -i eth0 -s 192.168.140.0/24 -j ACCEPT

# Allow anything in from insecure side destined for subnet
iptables -A INPUT -i eth1 -d 192.168.140.0/24 -j ACCEPT

# Allow anything from localhost to localhost
iptables -A input -i lo -j ACCEPT

# Forward traffic coming off the secure interface
iptables -A FORWARD -i ipsec0 -j ACCEPT

# Forward traffic going out the secure interface
iptables -A FORWARD -o ipsec0 -j ACCEPT

# Display Firewall
iptables -L -n -v

```

- Load the ruleset by executing this script: /etc/rc.d/ipsec.firewall

## Gateway ipsec setup:

We will now configure IPSEC on both the server and the client.

- On the gateway add a connection for each client to the /etc/ipsec.conf file

```
# at the beginning of the file change the ipsec line to match
interfaces="ipsec0=eth0"

# Client 1
conn client1
    # right = <IP of VPN gateway>
    right=192.168.140.250
    # Allow access out my default gateway
    rightsubnet=0.0.0.0/0
    # accept any connections from clients
    left=%any
    # use the public key associated with this FQDN
    leftid=@client1.wireless.netprl.calpoly.edu
    # use DNS to find public key
    lefttrsasigkey=%dns
    #Automatically allow this connection at startup
    auto=add
```

Start the ipsec service on the gateway

- echo 0 > /proc/sys/net/ipv4/conf/eth0/rp\_filter
- service ipsec start

## Client ipsec setup:

You will need to get the gateway's public key. To do this, run the following command on the gateway:

- ipsec showhostkey --right

Add the following connection to the client's /etc/ipsec.conf

```
conn warrior-to-gw
    # use the ip address of the interface of my default route
    left=%defaultroute
    # this is my unique ID
    leftid=@client1.wireless.netprl.calpoly.edu
    # IP of the VPN gateway
    right=192.168.140.250
    # use the ipsec as the default gateway
    rightsubnet=0.0.0.0/0
    # public rsa key of VPN gateway
    righttrsasigkey=0sAQNvLZankTF2zJt4rnJ2tUd3g/EmX1bD2oY03Q0g/pM4eA0
7K4/YA08H9FX9HqfiNdusQk7CLJIQ0csPfw2Riog3kK7WHMeqcoz+iDx3ynsP61zEss70s
/zP6txmqd/CAhdSrVyz0eE0QKC8XXEsAhW7LsI25Gx+Tj90HmtdWoabB/OVvfSx21+GNA
EO4pnXRfmIcckZT0YxvRmJ3J8URNXlFuY7/XLJlnKRWskMHGZHS40F+eS4G5j+cguvDI15
oLVSkCjmQ180tFvUQjFRo2DLgClu/LuImCVZi13LC1TqS82wAbp6cuv8otnV8DHJ1YTVu+
ylC6nT/dWE3G0dhGc60lbu3qyPr0Oy2TQptb8U+eT
    # add this connection at startup
    auto=add
```

Start the ipsec service on the each client:

- `echo 0 > /proc/sys/net/ipv4/conf/eth0/rp_filter`
- `service ipsec start`

Now, everything should be setup. In order to establish a connection to the server, run the following command on the client.

- `ipsec auto --up warrior-to-gw`

In order to make sure the packets are being encrypted, start up your favorite packet sniffer and sniff some traffic. You should be seeing all ESP packets. If not, something somewhere went wrong. Check Free S/WANs web site documentation.

## ***Conclusion***

Wireless security is a severe problem. There are new solutions emerging, including the new Wi-Fi Protected Access (WPA). This solution aims to replace the weak encryption of WEP. It will run on existing hardware and will strongly increase the level of data protection and access control [Grimm-2002]. However, why wait for this new protocol when a system has already been developed. A virtual private network can and does provide excellent security over wireless data links. VPNs were invented to carry private data over a public medium, and the 802.11b standard definitely makes the air a public medium.

The machine acting as the gateway has quite a few jobs. The test machine was a Pentium2 celeron 266. There was a slight decrease in performance, but we're sure this would be taken care of with a decent machine. We were able to surf the web and perform file transfers from a single client machine at an acceptable rate.

## ***References***

[Fluhrer - 2001] S. Fluhrer, I. Mantin, and A. Shamir. Weaknesses in the key scheduling algorithm of RC4. In Eighth Annual Workshop on Selected Areas in Cryptography, Toronto, Canada, Aug. 2001.

[Grimm – 2002] C. Brian Grimm. Wi-fi Protected Access – Overview. Wi-Fi Alliance. 2002 [http://www.wi-fi.org/OpenSection/pdf/Wi-Fi\\_Protected\\_Access\\_Overview.pdf](http://www.wi-fi.org/OpenSection/pdf/Wi-Fi_Protected_Access_Overview.pdf). Viewed Nov. 7, 2002.

[Borisov – 2001] N. Borisov, I. Goldberg, and D. Wagner. Intercepting mobile communications: The insecurity of 802.11. MOBICOM 2001 (2001).

S. Kent and R. Atkinson. Security Architecture for the Internet Protocol. RFC 2401. The Internet Society, November 1998. <http://www.rfc-editor.org>.

R. Thayer, N. Doraswamy, and R. Glenn. IP Security Document Roadmap. RFC 2411. The Internet Society, November 1998. <http://www.rfc-editor.org>.

D. Harkins, and D. Carrel. The Internet Key Exchange (IKE). RFC 2409. The Internet Society, November 1998. <http://www.rfc-editor.org>.

D. Maughan, M. Schertler, M. Schneider, and J. Turner. Internet Security Association and Key Management Protocol (ISAKMP). RFC 2408. The Internet Society, November 1998. <http://www.rfc-editor.org>.

S. Kent and R. Atkinson. IP Encapsulating Security Payload (ESP). RFC 2406. The Internet Society, November 1998. <http://www.rfc-editor.org>.