International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:5, No:5, 2011

A New Approach of Wireless Network Traffic on VPN

Amir Rashid, M. Saleem Khan, Freeha Zafar

***Abstract*—**This work presents a new approach of securing a wireless network. The configuration is focused on securing & Protecting wireless network traffic for a small network such as a home or dorm room. The security Mechanism provided both authentication, allowing only known authorized users access to the wireless network, and encryption, preventing anyone from reading the wireless traffic. The mentioned solution utilizes the open source free S/WAN software which implements the Internet Protocol Security –IPSEC. In addition to wireless components, wireless NIC in PC and wireless access point needs a machine running Linux to act as security gateway. While the current configuration assumes that the wireless PC clients are running Linux, Windows XP/VISTA/7 based machines equipped with VPN software which will allow to interface with this configuration.

***Keywords*—**Wireless network security, security network, authentication, encryption and internet protocol security.

I. Introduction

TODAY is the world of Wireless communication! The Wi-Fi mechanism has begun and there are access points popping up all over the globe. This new craze, based on the 802.11n standard, is the new hit among students and professionals everywhere. The ease of use due to wireless connectivity and parallel speed is appealing to all computer users. Almost all conferences everywhere are now advertising free, open wireless access to all attendees. Unfortunately, wireless access is easy to spread than contain. Anyone with a wireless accessibility can see the traffic and read the data. There have been more advances in wireless security, namely WEP (Wired Equivalency protocol) & WPA (Wi-Fi Protected Access); however these protocols have been proved to be less than adequate [1]. We all think as we send off our plain text passwords, hoping no one out there will notice.So, what can be used to secure wireless LANs from attackers? There must be a secure, private link to our access point so that no one else can read the actual data. The proposed suggestion is to implement a Virtual Private Network (VPN) over such insecure wireless link. VPNs establish an encrypted session for traffic to pass through between the client-server. VPNs provide authentication to ensure that only registered users use the wireless access point. This paper presents an easy method to implement, scalable, and free solution based on VPNs to secure wireless communications. A wireless network proposed in this paper refers to one or more computers, communicating via 802.11b standard infrastructure mode (access point). All clients must be MAC level authenticated and associated to the access point. Once this process has been done, the user is part of the wireless network and can send data to others on this network.Wireless network have many advantages over wired network. First, and the most obvious, is the fact that there are no cables to run. Wires cost accumulate rapidly and degrade the visual appearance of a home or office. Wireless access points are fairly cheap and most come with setup right out of the box. The comfort of use and affordability make this device a top choice among computing enthusiasts. Another contributing advantage of wireless networks is its speed. The 802.11b/g standard moves data at 54 MB/s, which is faster than most home Internet connections. The encryption mechanism 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 [2]. It was designed to provide the maximum security as a wired Ethernet connection, which provides no built in encryption. Since, the only security offered by wired network is the physical denial of access; WEP cannot be assumed as a means of guaranteeing secure data transmission. The speed at which 802.11b networks are spreading, something must be done to secure data traffic. Many universities and organizations are already implementing wireless LANs, both intentionally and unintentionally. Some wireless access points come out of the box with no encryption or restrictions on wireless access. This can result in a huge security lapse into a companies or universities backbone.A virtual private network is a mechanism of communicating data securely over an insecure network. A private session is established between a VPN enabled client and VPN gateway and all data transmitted across this connection are encrypted. Anyone monitoring the traffic between the client and the gateway can only see a standard IP packet header but the rest of the packet will be encrypted. The standard IPSec protocol is used to implement most VPNs as specified in RFC 2401 [3] and RFC 2411 [4].There are two different modes of VPNs: Transport mode and Tunnel mode. In Transport mode an application encrypts the traffic between a client and a server. An example of this is a secure connection establishment to a bank. Using tunnel mode means that any application not supporting encryption will send its traffic in form of clear text.In this proposed new approach of wireless network traffic on VPN, it is suggested to encrypt all traffic going across our insecure wireless link using VPN in tunnel mode.The frame work of this paper consists of: overview of the proposed system in section I, section II gives the design and implementation. Section III

Amir Rashid is with the MS(CS) Program in The Department of Computer Science, GC University, Lahore Pakistan
(e-mail: aamirrashid@live.com)
Dr.M. Saleem Khan is The Director of Computer Science Department, GC University, Lahore, Pakistan.
(email: mskgcu@yahoo.com)
Dr.Freeha Zafar is an Assistant Professor in the Department of Computer Science, GC University, Lahore, Pakistan (email:dr.f.zafar@gcu.edu.pk)

describes the system configuration. Section IV gives results and discussion, whereas section V gives conclusion and future work. Acknowledgement of the proposed work is in section VI.

II. Overview Of The VPN In Tunnel Mode

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

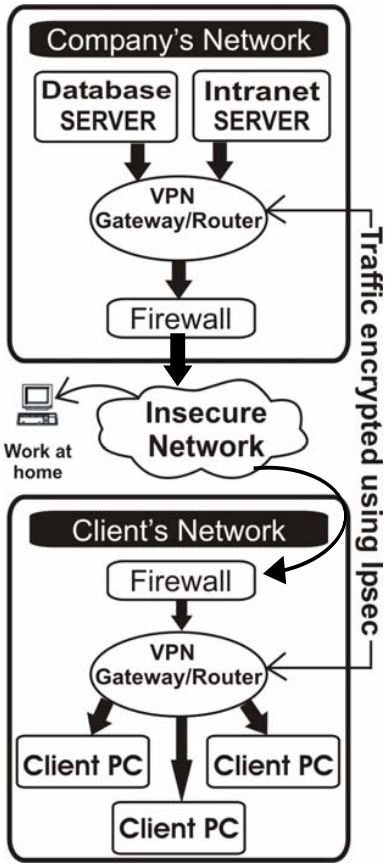

Fig. 1 The Structure of a VPN Used in Tunnel Mode

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

A. Overview of the Proposed System..

Figure 2 describes the structure of a VPN, securing wireless network. It depicts the scenario we are working with. In our scenario we have a number of Linux based clients who wish to communicate over an insecure wireless network. In addition to securing the 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 XP/Vista/7 that will work on our VPN configuration.

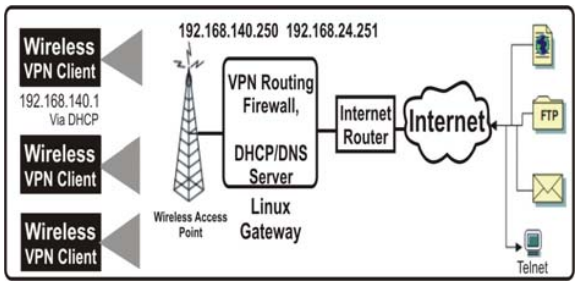

Figure 2. The Structure of Proposed Secure Wireless Network Using VPN

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

III. Design And Implementation

In order to support the proposed scenario the Linux machine must carry out a number of configurations.

A. Configurations

1. **DHCP Server** – The Linux machine is the DHCP (Dynamic Host Configuration Protocol) server for all of the wireless clients.

2. **DNS Server** – The DNS (Domain Name System) configuration will be used to provide the IPSEC key lookups.
3. **VPN Gateway** – The machine will be running the VPN software and is accountable for encrypting/decrypting traffic across the wireless network.
4. **Default Gateway/Router** – The Linux System will be configured to route traffic for the clients to the Internet.
5. **Firewall** – In order to avoid unauthorized users from accessing our network the Linux machine will be running firewall software (iptables).

B. Software overview

It is proposed to exercise Free S/WAN (pronounced free-swan), an IPSec implementation for the Linux operating system. Free S/WAN is freely accessible under the GPL license. The official web site is http://www.freeswan.org. This is an outstanding web site and will be able to answer almost any question you may have regarding the Free S/WAN package. Free S/WAN uses IPSec to encrypt traffic at the IP level of the protocol stack. This implementation uses (IKE) Internet Key Exchange as a way of securely exchanging keys. This exchange is based on the Diffie Hellman key exchange protocol. The particulars of the exchange will not be covered here and can be found in RFC 2409 [5]. IKE uses the (ISAKMP) Internet Security Association Key Management Protocol to send the keys back and forth across the network. This protocol is thorough in RFC 2408 [6]. In order to guarantee authentication, RIVEST SHAMIR ADLEMAN (RSA) keys are used in this exchange. These keys ensure much more security than shared passwords. The server will use the (DNS) Domain Name Service to authenticate the appropriate keys. Instead of a normal hostname lookup (type A), the server perform a key (type KEY) lookup to obtain a key for the client. Once a secure tunnel has been established, Free S/WAN uses an Encapsulated Security Payload (ESP) to transmit data. The particulars of ESP can be found in RFC 2406 [7]. All these technologies pooled together are used to create the encrypted tunnel for our data.

C. Proposed Scenario Server Configuration

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

These steps are:

1. *Install the Free S/WAN software on both the client and gateway Systems*

The gateway System we used is running Red hat Linux 9 with the 2.4.20 (default) kernel. The client machine is a laptop running Red hat Linux 7.2 with kernel 2.4.7-10 (default).
First download the two RPMs appropriate to the Systems. There is no separate RPM for free S/WAN clients and the gateway. (Note: these files can be found on the Free S/WAN website)

Red hat 9 with 2.4.20 kernel
(ftp://ftp.xs4all.nl/pub/crypto/freeswan/binaries/RedHat-RPMs/2.4.20-18.7/freeswan-module-2.06_204.20_18.7-0.i386.rpm)
(ftp://ftp.xs4all.nl/pub/crypto/freeswan/binaries/RedHat-RPMs/2.4.20-18.7/freeswan-userland-2.06_204.20_18.7-0.i386.rpm)
Red hat 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 Systems:
(RPM –ivh freeswan-module-*.rpm)
(RPM –ivh freeswan-*.rpm)

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

- Service IPSec stop (remember to stop this on both the client and the gateway)

2. *Configure DNS on the gateway System. This configuration will include both System names as well as the public keys needed for encryption*

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

- On your gateway Linux System 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 System, 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 require an addition of an entry for each wireless client that will be using the VPN. To do this run the following command on each client System:

(This assumes you have installed Free S/WAN on each client machine)

- IPSec showhostkey

This command will produce 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 System. (It is probably easiest to save the output of the "ipsec showhostkey" command to a file on the client system and then paste them into /var/named/wireless.netprl.calpoly.edu.zone on the gateway).
- Modify localhost.localdomain to a unique identifier. Here is an example for a client1.

```
client1  IN     KEY    0x4200 4 1
AQOVx0Jjl/XwiMVdyea8oUdRXDauxpp14pekpRa+m5FEu0k4bOkL2PEQU83rXtKlPaN286zX
DJCHl/Ik3PGc2sgA+tuXJ3Syq4a21rqGBnibTJRcbF/zYQDRtLegHnURw3qqXep83F2/s0YL
toeGerc7McMJKAE7De8CerOb8vsg/h6jBPmxQH8kjo167CKdKTFm0uMQAfLNCZJcOyxJmcyh
295vwdFFrwoDbOyrhn91l5DKP6+ZiRFIOIR6ItOOheSwrhyMSE7aFf9YgczaG48HV/2eaLdo
xTQOciFQ+hIxRudh8l4oCSPEvvC/2Ot6+3PKsHNmjmyO3BdJvNR/lhcin+ijFuBIrVIv49HB
hWo3rk6x
```

Now start the DNS server on the gateway System.

- service named start

Next edit the gateway System'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 show the RSA key for the client you just entered. If not, something went wrong above.

3. Configure the gateway System to be a DHCP server

We now need to set up the DHCP (Dynamic Host Configuration Protocol) server on the gateway. The DHCP server will configure IP addresses for all clients, default gateway and DNS server. The client systems will be on the 192.168.140.0 subnet. The wireless subnet is named: wireless.netprl.calpoly.edu. Systems on this subnet are named client1, client2, etc. Notice that the DNS server for the clients is not the gateway System. It is just a normal DNS server used for Internet connection (in our case 10.10.10.254 and 10.10.11.254).

- On the gateway System, 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;
}
```

4. Set up the gateway's firewall & routing

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

- On the gateway System, 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 rule set by executing this script: /etc/rc.d/ipsec.firewall

5. *Turn on the IPSEC services for the tunnel.*

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

- 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
  Leftrsasigkey=%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

D.Proposed 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 System /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
      rightrsasigkey=0sAQNvLZankTF2zJt4rnJ2tUd3g/EmX1bD2oY03QOg/pM4eA0
7K4/YAO8H9FX9HqfiNdusQk7CLJIQOcsPfW2Riog3kK7WHMeqcoz+iDx3ynsP61zEss70s
/zP6txmqqD/CAHdSrVyzOeEOQKC8XXEsAhW7LsI25Gx+Tj90HmtdWoabB/OVvfSx2l+GNA
EO4pnXRfmIcckZTOYxvRmJ3J8URNXlFuY7/XLJ1nKRWskMHGZHS4OF+eS4G5j+cguvDI15
oLVSkCjmQl8OtFvUQjFRo2DLgC1u/LuImCVZi13LClTqS82wAbp6cuv8otnV8DHJ1YTVu+
ylC6nT/dwE3G0dhGc6Olbu3qyPrOOy2TQptb8U+eT
 # add this connection at startup auto=add
```

Start the IPSec service on the each client System:

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

Now, everything should be setup and ready. In order to set up a connection to the server, run the following command on the client.

- ipsec auto --up warrior-to-gw

In order to ensure that the packets are being encrypted, start up your favorite packet sniffer and sniff some traffic. You should be able to see all ESP packets. If not, something somewhere went wrong during setup. For Support, check Free S/WANs web site documentation.

IV. RESULTS AND DISCUSSION

The security Mechanism provides authentication, allowing only known authorized users access to your wireless network, and encryption, preventing anyone from reading your wireless traffic and utilizes the open source free S/WAN software. It will run on existing hardware and will strongly increase the level of data protection and access control [8]. We are successful in transmitting wireless network traffic over unsecured public network securely.

V. CONCLUSION AND FUTURE WORK

Wireless security is a severe and huge problem. There are new solutions emerging, to provide more security to Wi-Fi networks. This solution aims to replace the weak encryption of WEP & WPA. However, why wait for these new Solutions when a system has already been developed. A VPN (virtual private network) can and does provide excellent security over wireless data links. VPNs were introduced to carry private data over a public medium, and the 802.11b standard definitely makes the air a public medium.

The System acting as the gateway has quite a few jobs. The test System was a normal Pentium 4 machine. There was a slight decrease in performance, but we're sure this would be taken care of with a decent System. We were able to surf the

web and perform file transfers from a single client system at an acceptable rate.

REFERENCES

[1] 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
[2] N. Borisov, I. Goldberg, and D.Wagner. Intercepting mobile communications: The insecurity of 802.11. MOBICOM 2001 (2001)
[3] S. Kent and R. Atkinson. Security Architecture for the Internet Protocol. RFC 2401. The Internet Society, November 1998. http://www.rfc-editor.org
[4] R. Thayer, N. Doraswamy, and R. Glenn. IP Security Document Roadmap. RFC 2411. The Internet Society, November 1998. http://www.rfc-editor.org
[5] D. Harkins, and D. Carrel. The Internet Key Exchange (IKE). RFC 2409. The Internet Society, November 1998. http://www.rfc-editor.org
[6] 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
[7] S. Kent and R. Atkinson. IP Encapsulating Security Payload (ESP). RFC 2406. The Internet Society, November 1998. http://www.rfc-editor.org
[8] C. Brian Grimm. Wi-fi Protected Access – Overview. Wi-Fi Alliance. 2002http://www.wi-fi.org/OpenSection/pdf/Wi-Fi_Protected_Access_Overview.pdf. Viewed Nov. 7, 2002