Goal: The goal of this lab is to use wireless tools such as Kismet and AirSnort to examine vulnerabilities that exist in wireless networks. We will see how we can sniff packets to obtain MAC addresses to get around filters and how to crack a WEP key.

Hardware: This lab will use three Pentium 4, 2.4GHz machines. Two will be wireless users running Windows XP and one will be an attacker running Linux 9.0. The wireless access point (AP) will be the D-Link tri-mode dualband wireless access point (DWL-7000AP), which means it can run 802.11a, 802.11b, and 802.11g, though we will only be using 802.11b for cracking. One Windows XP machine and the Linux machine use Linksys wireless A+G PCI adapter cards (WMP55AG). The other Windows XP machine (the client) uses a Linksys PCMCIA wireless network adapter, like those used with laptops, and Lucent PCI/PCMCIA bridges. You will not need your hard drive or any CD from the TA’s.

Software:
Ethereal
Kismet - [http://www.kismetwireless.net/](http://www.kismetwireless.net/)
Airsnort - [http://airsnort.shmoo.com](http://airsnort.shmoo.com)

Signup: There is only one setup for this lab so you will need to sign up in advance. Your group may sign up for a maximum of 2 hours in the future. Do not abuse this rule! Once you have used your “two hours” you may sign up again for another hour. You are never allowed to have more than 2 hours scheduled in the future. If you do not show up within the first 15 minutes, you forfeit your reservation and someone else may claim your reservation. The reservation sheet is located on the door of COC311. You do not have to demo anything to the TAs so you may work whenever you want.

Note on this format: Commands to be typed into a command line are in **bold** and preceded by a “#” sign, this is to be left out when typing the command. The setup for the wireless lab is shown on the next page. The monitor for each machine is labeled accordingly. WindowsXP1 will be used for the client (FTP and UDP) and WindowsXP2 will act as the server. Log onto the Linux box with username “root” and password “password”. If your Linux Machine is restarted, be sure to select the hostap kernel from
the initial menu and login with root and “password”.

Introduction

IEEE standard 802.11 was designed mainly to enhance a user’s accessibility to the Internet by allowing wireless capabilities. Today, with the growth of high-speed Internet access, wireless networks are gaining popularity among both business and home users. For the business IT employee, this translates into reduced time and cost of connecting a new user to the network.
The three wireless standards use different frequencies, but the 802.11b standard, used in this lab, uses radio waves that transmit in the 2.45 GHz ISM band for transfer speeds of up to 11 Mb/s. There are basically two different configurations for a wireless network. The first is called an ad-hoc network and is used when there is no access point to link all wireless cards together. This is very similar to a peer-to-peer network and all network cards communicate with each other directly. The other configuration is called an infrastructure network and has at least one access point. This access point, or AP, acts as a network hub to connect the wireless network to either another wireless network or to a wired network. Here, we will look at how wireless security is implemented in an infrastructure network using just one access point. We will also show how important it is to implement different wireless security measures to ensure the protection of your data.

Wireless Networks and the WEP

First, there are two basic types of attacks on a wireless network. The first one makes use of the broadcast properties of wireless technology to enter a network. This consists of being located close to a building with a wireless network employed and using a laptop, a wireless card, and a piece of software that can scan the wireless network. Of particular importance is the service set identifier, or SSID. This is the name of the wireless network and can be used to gain access if it is unsecured. If you cloak (turn beacons off) your SSID which means no one can see it by using an auto find of networks but you leave the default SSID unchanged, a hacker could try the common SSID and connect your network (assuming WEP is off). MAC address filtering can be used to increase the security of your network; however, MAC address can be spoofed which will then allow an attacker to gain access.

The second type of attack utilizes some of the vulnerabilities in the Wired Equivalent Privacy, or WEP, key. The WEP key utilizes an RC4 encryption algorithm, also known as a stream cipher. The sender takes a key and expands it to a lengthy random key stream and then XORs that with the information that is being sent. The receiver also has the same key and XORs the cipher-text, which gives the original information. This presents a problem because when an attacker has obtained two cipher-texts encrypted with the same WEP key, he can then XOR the two together and get the original information without needing to decrypt it with the WEP.

To prevent against this, an integrity check is implemented using an Initialization Vector, or IV. This vector prevents the same random key stream from encrypting two different packets. Unfortunately, it is only 24 bits long, which means that a busy access point will have to reuse the same random key stream sometime, usually within a few hours. An attacker will still be able to sniff this information off the network and use XOR to obtain the original information. These are just basic descriptions of the types of attacks an attacker will use to gain sensitive materials from a wireless network, so it is important to utilize different security standards to protect you from outside attacks.

Unencrypted Traffic
The first step in securing an 802.11 wireless network is making sure your data is being encrypted. Here, we will explore just how easy it is to compromise data that is being sent unencrypted or "in the clear." WindowsXP1 will be running a simple FTP server, while WindowsXP2 will be used to connect to that server. The Linux 9.0 machine is where we will sniff out useful data from the air and display it to the user.

It should be noted that the results of this experiment would remain the same whether it is a Telnet server being utilized, an FTP server, or any other service that does not encrypt its data before it is sent over the network. A telnet service runs by establishing a network connection between two hosts, and then simply transferring and then echoing every character the user happens to type on their keyboard. Thus, all our Linux 9.0 sniffer needs to do is listen to the packets and pick out the data portions in order to accomplish our goals at this point. We will do this first by using a packet sniffer we are already familiar with, Ethereal and then by looking at one designed for wireless sniffing called Kismet.

Before you begin, make sure the AP has been reset back to no MAC filtering and also no encrypted (WEP) settings. Do the following:

- Using a web-browser on WindowsXP2, connect to [http://192.168.1.144](http://192.168.1.144)
  - Username is “admin”. Password is “password”

To turn off the MAC control list filtering on the access point, connect to the AP:
- Click the advanced tab
- Choose enable MAC control filtering and select disable
- Click APPLY
- Wait for one minute as instructed and click OK

To turn off WEP on the Access Point:
- Connect to the AP again
- Click the “Advanced” tab on top then the “Encryption” tab on the left
- Make sure the “Disable” radio button next to WEP is clicked
- Click Apply, after you changes have been saved click continue

Make sure the last lab group turned off WEP on the two Windows XP machines. To do this on each of the two Windows XP machines:

- Start->Control Panel->Network and Internet Connections->Network Connections
- Right Click on Wireless Connection -> Properties->Wireless Networks tab
- Select wireless_dev from the Preferred networks box and click
properties
- Remove the check by the Data encryption (WEP) box
- Make sure the “The Key is provided for me automatically” is NOT checked
- Click OK
- Disable and enable the network device: Again right click on Wireless Connection -> Enable, then Wireless Connection -> Disable

Make sure that all three computers can communicate with the AP. In our case, this is accomplished by running on each of the three machines:

```
#ping 192.168.1.144
```

In the event the linux machine cannot ping, on the linux machine

```
#ifconfig ath0 down
#ifconfig ath0 -promisc
# ifconfig ath0 192.168.1.50
#ifconfig ath0 up
#iwpriv ath0 mode 2 //sets wireless device to listen on B (could use A,G)
#iwconfig ath0 essid wireless_dev
#iwconfig ath0 mode managed
#iwconfig ath0 channel 2

#iwconfig (you should see essid and mode set correctly)
#ifconfig (you should see wlan0 up with correct IP)
```

now try to ping the Access Point
```
#ping 192.168.1.144
```

This ends the procedure to reset the lab setup so you are starting at the proper configuration. Previous lab groups may have left the machine in another undesired state.

To start your lab, we are going to switch our Linux machine into monitor mode so that it can look at all the traffic flying by in the air and display the statistics it gathers to us.

**Exercise #1:** Use ethereal to capture ftp passwords. Ethereal is already installed on the linux machine. Use the following instructions to help you accomplish this:

On the linux machine we need to set the wireless card to promiscuous mode and in monitor mode. This is accomplished with
```bash
#ifconfig ath0 promisc
#ifconfig ath0 mode monitor
#ifconfig ath0 channel 2

Now start up ethereal
Click the icon next to the RedHat icon, looks like a large “e”

Capture
Start
Interface needs to be “ath0”
Update in real time
Automatic scrolling
OK

Now move to the WinXP2 on which the FTP server is installed and get the service running. To get it running:

• Start -> All Programs -> Cerberus FTP server -> Cerberus FTP server

This starts the FTP server.

A basic user account with which to connect from the other computer has already been set up for you. It is user “public” and password “password”. From the other Windows XP machine, connect to the FTP server by bringing up a command prompt (start->run->cmd ->enter) and typing:

```bash
#ftp 192.168.1.150
```

Enter login “public” and password “password”. Next type dir. This will generate a good deal of data passing back and forth that ethereal can pick up on. Type “quit”. Switch back to Linux click on stop in the capture window. Here we see one of the first issues of sniffing wirelessly, HUGE amounts of data. Our card captured every packet that went by, regardless of channel or destination, leading to a huge amount of data. To help you sort through all this you have been supplied with Display Filters, which ethereal uses to filter what you see.

• In the lower left hand corner of Ethereal click “Filters”
• Select either “wireless_dev (bssid)” or “ftp”

These will help you minimize the packets you need to search through to get useful information.

**Answer Questions at the end of this lab.**
MAC Control List Subversion

One way to restrict the access to a wireless network is to allow only certain MAC address to connect. Almost all access points have such a feature, and it theoretically should provide a decent level of security. However, these MAC addresses can easily be spoofed.

Exercise #2: Enable MAC address filtering on the access point and show that the attacker can gain access to the network by spoofing his address. Show screen captures and the like to show how you determined a valid MAC and how you gained access, include these with the questions section you turn in. Use the following instructions to help you accomplish this:

To enable MAC control list filtering on the access point, connect to the AP.

- using a web-browser on WindowsXP2, connect to http://192.168.1.144
- Click the advanced tab
- Click on the “Filters” tab on the left
- Change “Access Control” drop down box to “accept” for BOTH A & G (left and right columns)
- Place the MAC address of each Windows XP machine in the first two entries of the table. ASIDE: To determine the MAC address, open up a command window in Windows and type: `ipconfig /all`
  *Make sure you get the MAC of the wireless device
- Click “apply”, and after your changes have been saved message, “continue”
- On the two Windows XP machines, disable and enable the wireless connections to reset them. (see previous instructions on how to enable and disable the wireless cards in the Windows XP machines).

The Linux machine should not be able to connect to the wireless network now, to see this try to ping the router (however the Windows XP machines should be able to ping 192.168.1.144). The Linux machine can, however, still sniff the wireless network and see what MAC addresses are allowed on the network.

Let’s use Kismet on our Linux 8.0 machine to see the active MAC addresses on a network.

The Linux card should already be in monitor mode, but these commands will make sure:

```
# ifconfig ath0 promisc
# iwconfig ath0 mode monitor
# iwconfig ath0 channel 2
```

The Linux machine is now set up for our purposes but it should be noted that no normal network operations could be accomplished now that we are in monitor mode. Next run:
In a new window run:

`.kismet_client`

Finally, as you are monitoring, change the listening channel a few times by doing:

`.iwconfig ath0 channel 11` (or other numbers between 1-11)

This launches the kismet application itself and begins to display data. (Aside: kismet logs all packets it sees in a file which it stores in `/root/Kismet_logs`). The interface is entirely keyboard driven and if you get lost, you can display the help screen at any time by pressing `H`. Type in `H` and take a look at the commands. Use `+` to scroll down so you may read all the help info. Use `Q` to quit the help menu.

Several useful options are given but to begin with, simply hit `S` then `S` again in order to sort the wireless networks displayed by their SSID. We want to look at data coming from the network identified as `wireless_dev` so move down to it using the arrow keys and "tag" it by hitting `T`.

Now use the `C` option for the tagged wireless network. Do a ping from the Windows XP machines to generate some traffic from each wireless card and note how the client list in kismet shows you the client MAC address. If you are not able to see the MAC address with traffic generated by ping, use the UDP traffic generator discussed later in this lab to create traffic. (Some have reported that ping does not cause the MAC to show up and more traffic like the UDP traffic generator described later in the lab must be used).

Now that you're finished running kismet, it is important to shut it down properly or the wireless card will not return to normal operation. Do this by exiting out of the kismet application (use `Shift Q`) and then running the command in another terminal window:

```
# ifconfig ath0 –promisc
# iwconfig ath0 mode managed
# iwconfig ath0 channel 2
```

This places the wireless card back in its standard mode and it can now resume standard network operations.

Now lets spoof a MAC address (and an IP address as well) that is allowed on your network. We would need to find the IP addresses as well as the MAC addresses but for the purpose of this lab we will just cheat and look on the windows machine to find the IP address of that windows machine. Note we still have MAC address filtering turned on in the access point.

First lets change the IP address of the linux machine from 192.168.1.50 to that of the WindowsXP1 machine 192.168.1.100:
On the linux machine:

```
# ifconfig ath0 192.168.1.100
```

Now lets change the MAC address of the wireless card in the linux machine:

In a terminal window:

```
# ifconfig ath0 down
# ifconfig ath0 hw ether <MAC address obtained with Kismet>
# ifconfig ath0 up

# iwconfig ath0 essid wireless_dev(A)
# iwconfig ath0 mode managed
# iwconfig ath0 channel 2

# iwconfig (you should see essid and mode set correctly)
# ifconfig (you should see wlan0 up with correct IP)
```

Now notice you can ping the access point. You are using another machine’s MAC and IP address and you are allowed on the wireless network even though the MAC access control list is being used. It looks like the MAC access control list was a good idea but easily broken!

Now undo the nasty deed you have just committed by:

```
# ifconfig ath0 down
# ifconfig ath0 hw ether <original MAC address> (or just make up one)
# ifconfig ath0 up
# ifconfig ath0 192.168.1.50
# ifconfig (make sure original IP and MAC are back)

# iwconfig ath0 essid wireless_dev
# iwconfig ath0 mode managed
# iwconfig ath0 channel 2
# iwconfig (you should see essid and mode set correctly)
```

You should turn off MAC access list in the AP now by:

- Using a web-browser on WindowsXP2, connect to [http://192.168.1.144](http://192.168.1.144)
- Username is “admin”. Password is “password”

To turn off the MAC control list filtering on the access point, connect to the AP.

- Click the advanced tab
- Click on the “Filters” tab on the left
- Change the “Access Control” drop down to “disabled” on BOTH A & G (the left and right columns)
- Click “apply”, wait 1 minute, click “continue”

**Encrypted Traffic**

Data security in 802.11 is accomplished by what is known as Wireless Equivalent Privacy (WEP). The RC4 stream-cipher algorithm is used to encrypt the data. WEP relies on a secret key, normally 40 bits, and an initialization vector (IV), which is 24 bits, as a seed for the algorithm. The encryption of a frame proceeds as follows:

First an integrity checksum is calculated on the plaintext message and appended to the end to create the message we want to send.

\[ M = \{\text{Plaintext, Checksum}\} \]

The second stage involves encrypting the message \( M \) using the RC4 algorithm. This is done by choosing an IV and concatenating it with the secret key \( k \). The RC4 algorithm will generate a long sequence of pseudo-random bytes specific to this secret key and IV pair. This sequence is known as a keystream. To encrypt the packet, the plaintext message \( M \) is XORed with the keystream producing an encrypted message \( E \).

\[ E = M \oplus \text{RC4(IV,k)} \]

The encrypted message \( E \) is now ready to be transmitted. Figure 1 shows a graphical representation of what encryption looks like.

**Note:** This figure is taken from source [3]

![Diagram of encrypted WEP frame]

To decrypt a packet, the process is reversed. Once the packet is received, the IV used in
the encryption is retrieved from the unencrypted header. Using the IV, the keystream is regenerated using the secret key $k$ that both sides of the transfer have. The encrypted message is then XORed with the regenerated keystream, which will leave just the plaintext message that was initially encrypted.

$$M = E \text{xor} \ \text{RC4}(IV,k)$$

The checksum is then analyzed to ensure that the message has not been tampered with. Assuming everything checks out, the original plaintext message is passed along.

While WEP may sound like a great idea, it is inherently flawed on many levels. For the scope of this lab, the major flaws lie in the use of the initialization vector and the RC4 algorithm itself. In a paper entitled Weaknesses in the Key Scheduling Algorithm of RC4 by Fluhrer, Mantin, and Shamir, the authors propose a method in which under certain conditions the key setup algorithm of RC4 can leak information about the secret key. To attack RC4, they propose to search for specific IV’s that place keystream in this vulnerable state. In the parlance of this attack, these vectors are called “interesting”. By collecting enough of these “interesting packets” the entire secret key can be reconstructed.

This attack was first implemented in a paper called Using the Fluhrer, Mantin, and Shamir Attack to Break WEP, written by Stubblefield, Ioannidis, and Rubin. Using the techniques described in this paper, the programmers of two programs, WEPcrack and ArirSnort, released their own attacks to the internet soon after. In this lab, we will use AirSnort to break the WEP encryption and retrieve the shared key. Generally, AirSnort requires capturing between 5-10 million packets that on a busy network can be obtained in a few hours. A nice feature of AirSnort is its ability to save the current captured packets to a file and load them at some other time to continue trying to crack the key. This means that an attacker can work over many days to crack the encryption.

Here is what the AirSnort documentation says about what kind of packets it is looking for (source [5]):

Useful packets are those with the following property of their IV; the first byte is a number three greater than one of the offsets of the bytes of the key. For 128 bit encryption, this means a number from 3-16. The second byte must be 255 and the third byte can have any value. This means that for every byte of the key, there are 256 weak IVs. As of version 0.2.0, two additional classes of IVs are recognized by airsnort. See the source code for details.

Turning on WEP
First, we must enable WEP in the AP. To do this, do the following:

- Using a web-browser in WindowsXP2, connect to http://192.168.1.144
- User name is “admin” Password is “password”
- Click on the “Advanced” tab on top and the “Encryption” tab to
Click the “Enable” radio button next to WEP
- Change the Key length to 64bits
- Enter the WEP key, (We used the pass phrase ece4883 WEP key and this generated the one we are using for us
  - Use the key: \textbf{08905EF49F} in key 1
- Click apply and you will then no longer be able to communicate with the Access point from this PC until you turn on this PC’s WEP on its wireless card.

Now we must configure the Windows XP boxes to use WEP to connect to the AP
- Start->Control Panel->Network and Internet Connections->Network Connections
- Right Click on Wireless Connection -> Properties->Wireless Networks tab
- Select wireless_dev from the \textbf{Preferred networks} box and click properties
- Click the Data encryption (WEP) box
- Make sure the ”The Key is provided for me automatically” is NOT checked
- Select Hex and type the WEP key into the Network Key box \textbf{08905EF49F}
- Click OK and then OK again in the next area
- Disable and enable the XP wireless cards using the instructions given previously in the lab.

The Windows XP machines should now be able to connect to the network (ping 192.168.1.144) using WEP. The linux machine does not have WEP turned on so in theory it cannot see the wireless network.

\textbf{Using AirSnort to Break WEP}

\textbf{Exercise 3:} In this exercise we will run AirSnort on our Linux 8.0 machine in order to collect packets and crack the WEP key.

On the Linux machine, the wireless card should first be placed in monitor mode.

\begin{verbatim}
#ifconfig ath0 promisc
#iwconfig ath0 mode monitor
#iwconfig ath0 channel 2
\end{verbatim}

This will place have the card monitoring channel 2. If you wish to monitor on another channel, type iwconfig ath0 channel X. Also, you can use the iwpriv command to choose between A, B, and G. example iwpriv ath0 mode 3 chooses to monitor G (note: you
To run AirSnort, do the following in Linux:

```
# cd /root/Airsnort-0.2.1b/src
# ./airsnort
```

This will launch the AirSnort program. By selecting File->Load Crack File or File->Load pcap file, you are able to use previously captured traffic to try to crack the WEP key. For this lab, a crack file has been provided for you to speed up the recovery of the WEP key. Load the provided crack file by:

- Set it to monitor Channel 2 (not Scan)
- Set the device to ath0 (just type it in)
- Select “Other” Card Type drop down menu
- Set breadth to 4
- File->Load crack file
- Find the file named airsnort-crack-file located in the /root/WepLogs/ directory and double click on it

The crack file has already captured close to 5 million packets and is almost ready to recover the key. With the file loaded,

- Click “okay” to any warning
- Click the start button

Now on the two Windows XP machines, we will need to generate some encrypted traffic for airsnort to analyze in addition to this previously captured traffic on the lab network. A traffic generator client and server have been provided on both Windows XP machines for this purpose. (This code sends the user data aaaaaaaaaaaa, it is nice to know what the data is so that if you are running the UDP traffic generator when you collect some data with airsnort you will know where all the aaaa’s come from). This additional traffic along with the file that was provided for you will give airsnort enough traffic to crack the WEP in somewhere between minutes and hours depending upon the luck of the draw. (For example with just under 5 million packets sent and 2302 interesting packets among these, the key was cracked in just less than one hour). The more traffic you provide, the faster it cracks WEP. On WindowsXP1, run the udp traffic generator. On WindowsXP2 run the udp sink.

WindowsXP1
- open a command prompt
- # cd c:\wireless_lab
- # udp_gen 192.168.1.150 5555

WindowsXP2
- open a command prompt
- # cd c:\wireless_lab
• # udp_sink 5555

If you would like to speed up the process, you can start up a separate server/client pair using a different port. Also, you can send UDP traffic directly to the AP by pointing the generators at the IP 192.168.1.144. Furthermore, you can run continuing pings between the machines (ping –t destIP) The more the merrier. If you look at the activity light on the access point it should always be on now.

If you have not cracked the key after 100,000 packets or so, increase the breadth parameter every minute or two until it cracks. The breadth parameter controls how many guesses should be tried with a higher breadth parameter trying less likely guesses. Doing this several times over should eventually crack WEP. If you still have trouble, try reloading the crack file with a higher breadth.

When the key has been cracked, it will appear in the airsnort window under the "PW: Hex" and "PW: Ascii" tabs. (Recall in our case the key will be 08905EF49F). When you are finished, close Airsnort (no need to save this huge file) and then:

```
#ipconfig ath0 –promisc
#iwconfig ath0 mode managed
#iwconfig ath0 channel 2
```

Turn of the UDP traffic source and sink by using control C on the windows boxes in the windows with the programs running.

After cracking the key with Airsnort, answer the questions in the section provided.

**Exercise 4:** Decrypt the actual traffic on the wireless network and show how any evil hacker type can read the data. Note we are assuming we got the WEP key in exercise 3 and are using that answer here in exercise 4. Repeat exercise 1 with WEP turned on (as it is now assuming you are following the lab). After your FTP session file you created with kismet has been decrypted, open Ethereal and load the decrypted output file. The unencrypted packets are now available to be analyzed. Notice you can see the same stuff you saw before we turned on WEP. Use the instructions below.

In general, once the key is obtained, all previous traffic saved in logs can be decrypted and read. One method to prevent this is to change the key often, perhaps every 15 minutes or less. This is difficult since we have not seen an access point that would support this type of key switching.

AirSnort comes with a program to decrypt the traffic that you may have collected using kismet or AirSnort. To do this in Linux (wait not yet this is background):

```
• # cd /root/Airsnort-0.2.1b/src
• # ./decrypt -p <Key that airsnort cracked> -m <MAC address of access
```
point you wish to crack> -e <name of pcap log file containing encrypted packets> -d <name of pcap output log file>

As an example of the use of the decrypt program:

```
./decrypt -p 08:90:5E:F4:9F -m 00:05:5D:99:60:DB -e /root/Kismet_logs/<my file>.dump -d unencrypted.dump
```

Kismet automatically puts data it collects into /root/Kismet_logs/<todays date>.dump This is a binary pcap format file that ethereal can read.

To do exercise 4:

Start kismet using the instructions given previously
Run FTP between the Windows XP machines again as before
Shut down kismet as before, kismet automatically stores in /root/Kismet_logs directory the traffic it saw
Do an `ls -lart` to see your last file created with a “.dump” extension.
Run decrypt on your file (following the directions given)
Start ethereal and open your decrypted output file i.e. “unencrypted.dump”
Shut down kismet using the instruction given previously
Shut down the FTP server on the Windows XP machine

For more options about the program, just do `./decrypt` at the command line. For instance, a dictionary attack of several known keys can also be used against the encrypted log file in case one of those keys will work.

If you cannot get it to work on the standard Kismet Dump, try it on one of the other PCAP dumps in the /root/WepLogs folder (the goodPCAPcrack file is known to work although it is rather large… >2GB!)

There are no questions for this section, but include screenshots of both encrypted and unencrypted packets.

**To finish up the lab exercises and to be a good citizen, please turn off WEP by:**

Make sure the AP has been reset back to no MAC filtering and also no encrypted (WEP) settings. Do the following:

- Using a web-browser on WindowsXP1, connect to [http://192.168.1.144](http://192.168.1.144)
- Username is “admin” Password is “password”

To turn off the MAC control list filtering on the access point, connect to the AP.
- Click the advanced tab
• Click on the “Filters” tab on the left
• Change “Access Control” drop down box to “disable”
• Click “apply”, wait one minute and click “continue”

To turn off WEP on the Access Point:
• Connect to the AP again
• Click the “Advanced” tab on top then the “Encryption” tab on the left
• Make sure the “Disable” radio button next to WEP is clicked
• Click Apply, after you changes have been saved click continue

Make sure you turn off WEP on the two Windows XP machines. To do this on each of the two XP machines:

• Start->Control Panel->Network and Internet Connections->Network Connections
• Right Click on Wireless Connection -> Properties->Wireless Networks tab
• Select wireless_dev from the Preferred networks box and click properties
• Remove the check by the Data encryption (WEP) box
• Make sure the "The Key is provided for me automatically" is NOT checked
• Click OK
• Disable and enable the network device: Again right click on Wireless Connection -> Enable, then Wireless Connection -> Disable

Make sure that all three computers can communicate with the AP. In our case, this is accomplished by running on each of the three machines:

```
#ping 192.168.1.144
```

In the event the linux machine cannot ping, on the linux machine

```
#ifconfig ath0 –promisc
#iwconfig ath0 mode managed
#ifconfig ath0 192.168.1.50
#iwconfig ath0 essid wireless_dev
#iwconfig ath0 channel 2

#iwconfig (you should see essid and mode set correctly
#ifconfig (you should see wlan0 up with correct IP)
```
Sources:


Questions For Lab 10

Exercise #1:

1.1) What does each of the filters let through? What is the significance of the Hex number in the “wireless_dev (bssid)” filter?

1.2) What kind of data were you able to see (like passwords!)?

1.3) What are some ways to prevent this kind of packet sniffing?

1.4) What are the major disadvantages of leaving the traffic unencrypted?

Exercise #2:

No questions for this part, just include screenshots and explanations of them at the end of these questions.

Exercise #3:

3.1) What was the value of the broken key?

3.2) How many “interesting” packets did it take?

3.3) Explain why AirSnort is one of the most devastating attacks against 802.11 and
some possible ways to correct it.

Exercise #4:

No questions for this part, just include screen shots of encrypted and unencrypted packets.

How long did it take you to complete this lab? Was it an appropriate length lab?

What corrections and or improvements do you suggest for this lab? Please be very specific and if you add new material give the exact wording and instructions you would give to future students in the new lab handout. You may cross out and edit the text of the lab on previous pages to make corrections/suggestions. Note that part of your lab grade is what improvements you make to this lab.